



Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Mestrado em Engenharia Informática

Interfaces Persuasivas e Tangíveis

Miguel Ângelo Fernandes Almeida
(aluno n.º 26816)

1º Semestre de 2009/2010
22 de Fevereiro de 2010



Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Interfaces Persuasivas e Tangíveis

Miguel Ângelo Fernandes Almeida
(aluno n.º 26816)

Orientadora: Prof. Doutora Teresa Isabel Romão

Trabalho apresentado no âmbito do Mestrado em Engenharia Informática, como requisito parcial para obtenção do grau de Mestre em Engenharia Informática.

1º Semestre de 2009/2010

22 de Fevereiro de 2010

Agradecimentos

Em primeiro lugar quero agradecer à professora Teresa Romão, pela sua excelente orientação e por toda a ajuda preciosa prestada durante o desenvolvimento deste trabalho. Quero também demonstrar os meus agradecimentos ao colega Pedro Lobo, pela ajuda inicial que foi motivadora para a elaboração desta tese. À minha família e amigos, em especial à minha irmã Ana Rita Almeida, por todo o suporte que me foi oferecido ao longo deste ano.

Não posso esquecer de mencionar Teresa Riso e Sofia Lopes, por todo o empenho e ajuda prestada no desenvolvimento do caso de estudo. Por fim à direcção do Colégio Atlântico pela disponibilidade oferecida ao longo dos últimos dois anos, e pela cedência do espaço para a realização do caso de estudo.

A todos os meus mais sinceros agradecimentos.

Resumo

Com o avanço tecnológico o mundo real encontra-se cada vez mais repleto de sistemas computacionais que invadem o quotidiano, na tentativa de facilitar as tarefas humanas. Cada aplicação possui um modelo específico de interacção, que promove a necessidade de cada utilizador passar por uma fase de aprendizagem antes do uso da mesma. No futuro espera-se que esses modelos se tornem transparentes, proporcionando ao utilizador uma interacção natural com o sistema. Desta forma é possível aproveitar esses sistemas para consciencializar alguns membros da sociedade, sobre alguns comportamentos, de maneira a aproveitar os efeitos persuasivos e de aprendizagem que podem proporcionar. Nos dias de hoje, cada vez mais, as interfaces tangíveis se impõem, nas aplicações mais simples, sobre as interfaces gráficas convencionais, devido à sua facilidade de uso.

Esta dissertação tem como objectivo principal estudar comparativamente as interfaces gráficas convencionais e tangíveis, quanto às potencialidades persuasivas e de aprendizagem de cada uma. Para tal, foi estendido o desenvolvido de uma *Framework* que permite criar aplicações não intrusivas de carácter persuasivo, que através de consciencialização dos utilizadores, em relação aos problemas ambientais, os levam a adoptar comportamentos mais adequados à resolução desses mesmos problemas. Estas aplicações permitem detectar as condições do meio envolvente e as acções dos utilizadores e responderem de acordo com elas. Foi desenvolvida, também, uma *Authoring Tool* para auxiliar na criação de sistemas persuasivos e cientes do contexto envolvente do meio, por parte de utilizadores sem conhecimentos de programação.

Além disto, foi também construído um protótipo de sistema tangível como caso de estudo, o *Ecosystem Room*. Com este protótipo foram efectuados testes com crianças, que tiveram como objectivo principal a avaliação da usabilidade da interface, o seu potencial de persuasão e capacidade de instruir para valores e/ou matérias desconhecidas. São também descritos os resultados desses mesmos testes assim como os aspectos idiossincrásicos do sistema. Os resultados têm como objectivo elucidar para uma melhor compreensão do uso da tecnologia tangível, nomeadamente na aprendizagem das crianças.

viii

Palavras-chave: Interfaces tangíveis, Sistemas Persuasivos, *Feedback*, Sistemas Ubíquos, Interação com crianças

Abstract

With the technological advance, the real world is full of computer systems, which invade our everyday life in the attempt to facilitate humans tasks. Each application has a specific interaction model, so users need to go through a learning phase before they can use it. In the future, it is expected that these models become transparent, promoting a natural interaction between the user and the system. Therefore, it is possible to explore the persuasive effect these systems can provide, motivating users to adopt a certain attitude or behavior. Nowadays, tangible interfaces are starting to replace conventional graphical interfaces in several simple applications, due to its easiness of use.

The main objective of this master thesis is to analyze the persuasive potential of tangible interfaces when compared to conventional graphical interfaces. In order to do so a framework was developed allowing the creation of persuasive and non intrusive applications, through awareness of users, in relation to environmental problems, lead them to behave more appropriate to the resolution of those problems. These applications can detect the conditions of the environment and the user's actions and respond accordingly. It was also developed an Authoring tool to assist the creation of context aware persuasive systems, by users without programming skills.

Moreover, a prototype system was developed, *Ecosystem Room*, considered to be proof of concept for this master thesis. With this particular prototype same tests were performed with children, so as to evaluate the usability of the interface, its persuasive potential and its capacity to teach values and/or unknown subjects. The results of these tests and the system's properties are revealed, with the purpose to elucidate for a better understanding in the use of tangible technology, mainly concerning children's learning.

Keywords: Tangible interfaces, Persuasive systems, Feedback, Ubiquitous Systems, Children interaction

Conteúdo

1	Introdução	1
1.1	Introdução geral ou Motivação	1
1.2	Descrição e contexto	2
1.3	Solução apresentada	3
1.4	Principais contribuições	4
1.5	Estrutura da Tese	5
2	Trabalho relacionado	7
2.1	Interacção Pessoa-Máquina	7
2.1.1	<i>Feedback</i> na Interacção Humana	9
2.1.2	Tipos de <i>Feedback</i>	10
2.2	Sistemas Ubíquos	12
2.3	Sistemas Persuasivos	14
2.3.1	Objectivos	14
2.3.2	Métodos de persuasão	14
2.3.3	Exemplos de sistemas persuasivos	16
2.3.3.1	Facebook e Mixi	16
2.3.3.2	<i>Personal Health Information</i> (PHI)	18
2.3.3.3	<i>Calorie-aware Kitchen</i>	20
2.4	Interfaces Tangíveis	22
2.4.1	Exemplos de Sistemas Tangíveis	24
2.4.1.1	Tangible Newspaper	24
2.4.1.2	Cube	25
2.4.1.3	<i>Sensetable</i>	26
2.4.1.4	<i>Reactable</i>	27
2.5	Interacção com crianças	30
2.6	Modelo Predador - Presa	34
3	Metodologia e Desenvolvimento da <i>Framework</i>	37
3.1	Arquitectura da <i>Framework</i>	38
3.2	Arquitectura da <i>Authoring Tool</i>	39

3.3	Diagrama de classes	41
3.3.1	<i>Framework</i>	41
3.3.2	<i>Authoring Tool</i>	43
4	Descrição da implementação da <i>Framework</i>	45
4.1	Estruturas de comunicação	45
4.2	Implementação da <i>Framework</i>	50
4.2.1	<i>NewSensor</i>	50
4.2.2	<i>Context Processor</i>	51
4.3	Implementação da <i>Authoring Tool</i>	51
4.3.1	Wizard	52
4.3.2	Elementos gráficos	53
4.4	Implementação de módulos auxiliares	54
4.4.1	Simulador	54
4.4.2	Reconhecimento Facial	55
4.4.3	Leitura de <i>Data Matrix</i>	55
4.4.4	Girador e Sensores	55
4.4.5	<i>TextToSpeech</i>	56
5	Avaliação do Caso de Estudo	57
5.1	Desenvolvimento do caso de estudo	58
5.1.1	Arquitectura do <i>Ecosystem Room</i>	58
5.1.2	Regras e procedimentos	61
5.2	Avaliação da <i>Framework</i> e respectiva <i>Authoring Tool</i>	63
5.2.1	Participantes	63
5.2.2	Metodologia	63
5.2.3	Apresentação de resultados	64
5.3	Avaliação do <i>Ecosystem Room</i>	65
5.3.1	Participantes	65
5.3.2	Metodologia	66
5.3.3	Apresentação de resultados	67
5.3.4	Considerações globais	70

6	Conclusão e trabalho futuro	73
6.1	Conclusão	73
6.2	Trabalho futuro	75
A	Apêndice A, Manual de utilizador de <i>Authoring Tool</i>	77
A.1	<i>Editor</i>	77
A.2	<i>Wizard</i>	78

Lista de Figuras

2.1	Modelos conceptuais de Norman	7
2.2	Demonstração de HeatSink, retirado de [7]	10
2.3	Exemplo de <i>AwareKitchen</i> , retirado de [12]	13
2.4	Interligação de tipos de persuasão, adaptado de [8]	15
2.5	Exemplo de interface PHI, retirado de [20]	18
2.6	Sistema de Calorie-aware Kitchen, retirado de [3]	21
2.7	Arquitectura de uma GUI	23
2.8	Arquitectura de uma TUI	23
2.9	Mapeamento de GUI para elementos TUI, retirado de [10]	24
2.10	Exemplo de <i>Data Matrix</i> para o Jornal Tangível, retirado de [24]	25
2.11	Exemplo de cubo como interface Tangível, retirado de [24]	26
2.12	Exemplo de <i>Sensetable</i> , retirado de [21]	27
2.13	Arquitectura de <i>Sensetable</i> , retirado de [21]	27
2.14	Arquitectura de <i>Reactable</i> , retirado de [11]	28
2.15	Exemplo de <i>Smileyometer</i> , retirado de [22]	31
2.16	Exemplo de <i>Fun Sorter</i> , retirado de [22]	31
2.17	Exemplo de <i>Again Again table</i> , retirado de [22]	32
2.18	Equações representativas do modelo de Lotka-Volterra	34
2.19	Excerto do ficheiro Excel que simula as interacções predador - presa, segundo o modelo de Lokta-Volterra	35
3.1	Arquitectura da <i>Framework 1.0</i>	38
3.2	Arquitectura da nova <i>Framework</i>	38
3.3	Arquitectura da <i>Authoring Tool</i>	40
3.4	Diagrama de classes da <i>Framework</i>	41
3.5	Diagrama de classes da <i>Authoring Tool</i>	43
4.1	Estrutura do XML Schema para os ficheiros que descrevem as propriedades de um sensor	46
4.2	Estrutura do XML Schema dos ficheiros de comunicação entre os sensores e a <i>Framework</i>	47
4.3	Esquema de comunicação através dos ficheiros <i>InOut</i>	47

4.4	Estrutura do XML Schema dos ficheiros que listam os sensores adicionados à <i>Framework</i>	47
4.5	Estrutura do XML Schema, dos ficheiros XML que representam os objectos a desenhar na <i>Authoring Tool</i>	48
4.6	Estrutura do XML Schema para os ficheiros que configuram as acções de cada estado	49
4.7	Exemplo de configuração através da classe <i>Wizard</i>	52
4.8	Modelo de aplicação desenvolvido na <i>Authoring Tool</i>	54
5.1	Ambiente de <i>Ecosystem Room</i>	59
5.2	Mesa de <i>Ecosystem Room</i>	59
5.3	Criança a interagir com <i>Ecosystem Room Table</i>	59
5.4	Visão inferior da <i>Ecosystem Table</i>	60
5.5	Visão da interface gráfica convencional	61
5.6	Distribuição da idade dos utilizadores; média = 8.83; desvio padrão = 0.69	66
5.7	Distribuição do sexo dos utilizadores	66
5.8	Número de utilizadores que sabem ou não o que é um ecossistema	67
5.9	Número de utilizadores que sabem ou não o que é uma espécie em vias de extinção	68
5.10	Causas que podem provocar a extinção de animais	68
5.11	Número de animais que os utilizadores conhecem e que estão em vias de extinção	68
5.12	Motivação dos utilizadores antes das interacções; média=4.53; desvio padrão = 0.5	68
5.13	Satisfação do utilizador em relação ao jogo; média = 4.47; desvio padrão = 0.68	69
5.14	Satisfação do utilizador após ter jogado; média = 4.67; desvio padrão = 0.48	69
5.15	Preferência da interface, em relação ao jogo	69
5.16	Interface preferida para outros contextos educativos	69
5.17	Resultados da identificação do utilizador; média = 3; desvio padrão = 0	70
5.18	Resultados da compreensão do jogo/interacção; média = 2.13; desvio padrão = 0.57	70
5.19	Resultados da interacção com a mesa durante o jogo; média = 2.93; desvio padrão = 0.25	70
A.1	Exemplo de um grafo	77
A.2	Menu e submenus da <i>Authoring Tool</i>	77

A.3	Atribuir um nome ao estado	78
A.4	Início de <i>wizard</i>	78
A.5	Atribuir acção pré-definida ao estado	79
A.6	Seleccionar a acção interna	79
A.7	Seleccionar um sensor e respectivo método	80
A.8	Configurar <i>inputs</i>	80
A.9	Seleccionar próximo estado a ser executado	80
A.10	Seleccionar destino dos <i>outputs</i>	80
A.11	Afectar valores de <i>outputs</i> às variáveis	81
A.12	Configuração directa dos <i>outputs</i> provenientes dos sensores	81
A.13	Comparação dos valores das variáveis	81

Lista de Tabelas

2.1	Estratégias de persuasão de PHI, adaptado de [20]	19
2.2	Diferenças nas características de implementação, adaptado de [1]	33
5.1	Mapeamento das ameaças com os continentes	62

1 . Introdução

1.1 Introdução geral ou Motivação

Nos tempos correntes, os computadores são cada vez mais utilizados como meios de comunicação e aprendizagem. A capacidade de processamento dos dispositivos electrónicos cresceu tanto que a sua utilização passou, de meras operações de cálculos, para a complexidade de sistemas operativos robustos e com suporte a ambiente gráfico. Toda esta evolução abre novas portas a estudos sobre as tecnologias contemporâneas. Entre elas temos as tecnologias persuasivas e as interfaces tangíveis. Nestes estudos, o *feedback*¹ será um dos temas a ser abordado, uma vez que pode ser determinante para a capacidade de alteração da atitude e comportamento humano, reflectido pelas tecnologias acima referidas.

Com o aparecimento cada vez mais assíduo de novas tecnologias, as interações diárias com estas tornam-se também mais frequentes. Assim, é natural que sistemas que detectem algumas características do comportamento humano, no contexto social ou espacial em que são inseridos, sejam utilizados para interagir com o público, cativando-o para o objectivo pretendido. No paradigma das interfaces persuasivas, por exemplo, estas pretendem incentivar o utilizador a efectuar uma acção ou a ter um comportamento diferente do seu habitual, alterando-o na sua vida tradicional.

A existência de objectos físicos, capazes de comunicar com sistemas computacionais, fazem com que a interacção pessoa-máquina se possa tornar mais natural, de forma que ao manipular um objecto conhecido, ainda que simbólico, haja um mapeamento directo numa acção do sistema. Assim, se traduzem as interfaces tangíveis. Num mundo que se vê invadido por sistemas ubíquos, apresentados na secção 2.2, objectos do quotidiano, manipuláveis, podem ser a forma mais imediata de se conseguir uma interacção natural com um sistema. De notar o facto de que o conhecimento prévio do objecto por parte do utilizador é, por si só, uma vantagem na adaptabilidade deste ao sistema que pretender utilizar.

Várias interfaces possuem diferentes formas de interacção com o utilizador. Sendo o desafio de todas elas conseguirem mostrar ao utilizador uma resposta concreta e perceptível da acção que este executou, assim como um *feedback* correcto e atempado. No estudo de sistemas persuasivos, pretende-se avaliar se uma interface é ou não capaz de ensinar a um utilizador conceitos

¹resposta a uma instrução

que este não possuía, de forma não intrusiva, verificando *a posteriori* se foi possível alterar a atitude ou o comportamento do mesmo.

Com o agravamento das condições do meio ambiente e a crescente necessidade de evitar a degradação da natureza, cada vez mais as organizações ambientais apelam para que as pessoas alterem o seu comportamento ambiental. Desta forma, as crianças, que ainda se encontram em fase de aprendizagem, tendo então uma maior facilidade de alterar rotinas comportamentais, formam um público-alvo muito atraente para o investimento no ensino dos benefícios de boas práticas ambientais. Assim, o estudo de sistemas persuasivos torna-se importante na medida em que estes poderão consciencializar de formas menos intrusivas e mais apelativas. Com o aparecimento de interfaces tangíveis, torna-se possível influenciar o comportamento do utilizador através de objectos que este manipula no seu dia-a-dia.

1.2 Descrição e contexto

Os Humanos são seres de hábitos e rotinas. A alteração do comportamento adoptado por cada um é uma tarefa difícil, visto a reacção ser variável de pessoa para pessoa. A consciencialização dos problemas ambientais, como a preservação das espécies, é uma tarefa que necessita de ser inculcada a todos os seres Humanos. As tecnologias persuasivas têm recentemente vindo a ser estudadas e utilizadas de modo a possibilitar a desejada alteração de comportamento. As interfaces tangíveis são potenciais formas de persuadir, devido ao facto de poderem ser facilmente compreendidas pelos utilizadores e mais apelativas para algumas faixas de população, por exemplo, as crianças. Desta forma, levanta-se a questão da potencialidade das interfaces tangíveis na persuasão face às interfaces gráficas convencionais, que utilizam os periféricos como meios intermédios de interacção, como o rato, o teclado ou o monitor.

A *Framework* [14] desenvolvida no âmbito do trabalho de Pedro Lobo [15], designada por versão 1.0, inclui uma *Authoring Tool* e permite a um utilizador, mesmo com pouca experiência em programação, criar as suas aplicações persuasivas. Esta *Framework* apresenta ainda algumas limitações nas abordagens ao utilizador. Por exemplo, a sua implementação permite somente o desenvolvimento de aplicações que incluam um conjunto limitado de sensores, como por exemplo, infravermelhos e câmara de vídeo para detecção de códigos *Data Matrix*. Existia a necessidade de tornar esta *Framework* numa ferramenta mais versátil e estudar a sua capacidade, de forma a permitir o desenvolvimento de aplicações mais complexas e variadas, que

incluam novas formas de interacção, nomeadamente a inclusão do suporte à construção de interfaces tangíveis. Existia também a possibilidade de melhoramento de toda a interface gráfica apresentada pela *Authoring Tool*.

Este trabalho implementa esses melhoramentos e analisa as diferenças entre as interfaces gráficas convencionais e as interfaces tangíveis, no que se refere a efeitos persuasivos e de aprendizagem. Neste sentido, estuda-se qual a interface que mostra ser mais adequada para efeitos de persuasão e aprendizagem. Ou seja, a interface que se relaciona com o utilizador de uma forma mais adequada para conseguir consciencializar e alterar os comportamentos do mesmo. Para tal, foi alterada uma *Framework* [14], que possibilita a integração dos tipos de interfaces acima referidos e que permite o desenvolvimento de aplicações interactivas de carácter persuasivo para uso na aprendizagem de conceitos.

Esta tese de mestrado foi desenvolvida no âmbito do projecto **DEAP**² (PTDC/AAC-AMB/104834/2008). O trabalho precedente foi (e está a ser) realizado no âmbito da tese de Doutoramento de Joana Hipólito relativa ao *EcoDesign*(primeiros desenvolvimentos publicados na *persuasive 2007* [23]) e da tese e Mestrado de Pedro Lobo [15].

1.3 Solução apresentada

A solução dos problemas apresentados na secção anterior passaram pela remodelação da *Framework* referida e da respectiva *Authoring Tool*, de modo a que esta se tornasse mais robusta e abrangente na construção de ambientes persuasivos. Assim, a remodelação passou pela implementação dos seguintes requisitos:

- **Generalização da *Framework***, desta forma foi desenvolvido um método que possibilita a adição de novos sensores, algoritmos e mecanismos, incluindo o suporte à implementação de interfaces tangíveis. Método este que é apresentado em maior detalhe na secção 4.
- **Adição de reconhecimento facial**, o reconhecimento de um utilizador, que era efectuado através de reconhecimento de códigos *Data Matrix*, foi alterado para um módulo de reconhecimento facial. Este permite identificar um utilizador, após a detecção da sua presença, numa zona do sistema facilitando e personalizando assim a interacção com o utilizador.

²Developing Environmental for Awareness with Persuasive systems

- **Reformulação da interface gráfica da *Authoring Tool***, de modo a tornar mais intuitivo o uso da ferramenta, permitindo ao utilizador verificar o sistema que está a construir, no momento da construção, assim como visualizar o grafo de acções que poderão ser desencadeadas ao longo da interacção com o sistema.

Como processo de validação foi desenvolvido um sistema designado de *Ecosystem Room*. Este é vocacionado para crianças e tem como principais objectivos, dar a conhecer o funcionamento de um ecossistema e o impacto das nossas acções no meio ambiente, com vista à alteração do comportamento dos utilizadores face ao equilíbrio do ecossistema. Desta forma, é ensinado ao utilizador as relações entre os diversos componentes do ecossistema, assim como as ameaças que estão inerentes no mesmo. A construção deste sistema através da utilização da *Framework* e respectiva *Authoring Tool* permite testar todos os elementos de interacção do mesmo. Foi ainda efectuado um estudo comparativo de duas interfaces, uma interface tangível e uma interface gráfica convencional, na perspectiva de ser possível avaliar qual a interface que contribui, de melhor forma, para a aprendizagem e a alteração do comportamento de um utilizador.

1.4 Principais contribuições

Considerando as soluções apresentadas na secção 1.3, as principais contribuições que este trabalho fornece são:

- Desenvolvimento de uma *Framework* genérica, que permite a inclusão de qualquer sensor físico, algoritmo ou mecanismo no sistema a desenvolver, assim como vários tipos de interface. O que permite oferecer ao utilizador um conjunto de funcionalidades, a ser utilizado num sistema, muito mais vasto.
- Estudo da contribuição das interfaces tangíveis, no efeito de aprendizagem e persuasão de sistemas computacionais, tendo como base de comparação as interfaces gráficas convencionais.
- Nova visão de interacção persuasiva, utilizando interfaces tangíveis como meio de persuasão.

- Implementação de um sistema persuasivo tangível, destinado a crianças, onde se conscientizam os utilizadores para os impactos ambientais provocados pela sua acção comportamental.

1.5 Estrutura da Tese

A continuação desta tese possui a seguinte estruturação:

- Capítulo 2 (Trabalho Relacionado), este capítulo contextualiza vários aspectos que são utilizados ao longo de toda a tese, assim como refere estudos efectuados anteriormente e que podem estar ligados com o desenvolvimento deste trabalho.
- Capítulo 3 (Metodologia e Desenvolvimento da *Framework*), ao longo deste capítulo é apresentada a metodologia seguida no desenvolvimento da *Framework*. São também enunciadas as alterações à arquitectura da versão 1.0 da mesma.
- Capítulo 4 (Implementação do sistema), apresenta as alterações efectuadas à *Framework* já existente e descreve os aspectos técnicos de toda a implementação.
- Capítulo 5 (Caso de Estudo), descreve todo o sistema utilizado no caso de estudo, assim como os recursos que proporcionaram o mesmo. São também apresentados todos os resultados obtidos através dos testes efectuados, tanto para a *Authoring Tool* como para o protótipo desenvolvido.
- Capítulos 6 (Conclusão da elaboração da dissertação e trabalho futuro), neste capítulo são apresentadas as conclusões finais do estudo, assim como orientações para trabalho futuro que possa vir a ser desenvolvido.

2. Trabalho relacionado

2.1 Interação Pessoa-Máquina

Hoje em dia as tecnologias computacionais envolvem cada vez mais o quotidiano do ser Humano. A interação do Homem com essas tecnologias, e o estudo dessa mesma interação, denomina-se de HCI ¹ ou Interação Pessoa-Máquina . Desta forma HCI, pretende estudar a melhor forma de integrar e estabelecer a comunicação entre ambos, Homem e Máquina, através do conhecimento das duas partes, como por exemplo, as limitações computacionais, existentes nas máquinas e a problemática psicossociológica do ser Humano.

Para Norman Donald [19], a interação pessoa-máquina rege-se por modelos conceptuais desenvolvidos pelas entidades envolvidas presentes na interação, ou seja, a interface, o sistema, o utilizador e o designer do sistema, tal como pode ser observado na figura 2.1.

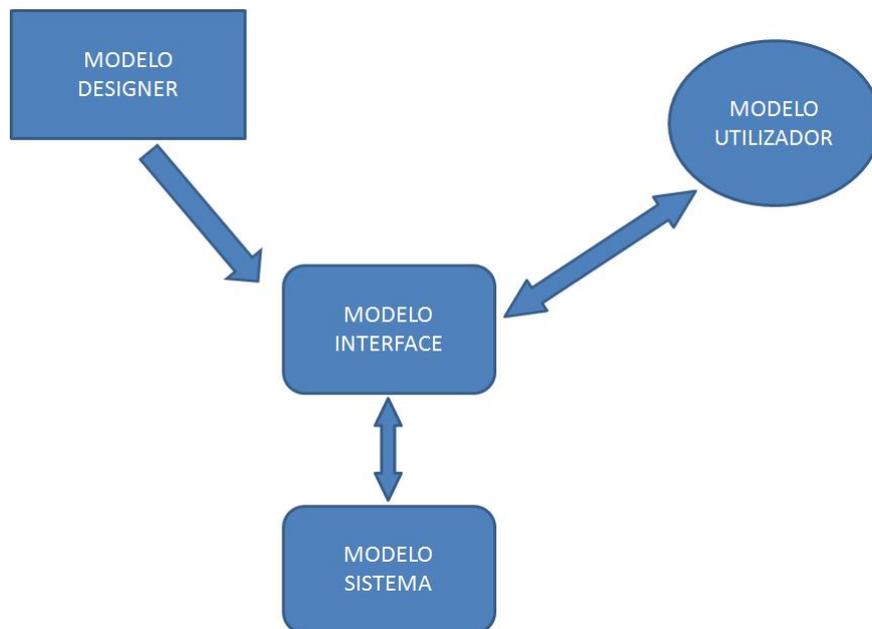


Figura 2.1 Modelos conceptuais de Norman

Como tal, é possível descrever cada elemento da seguinte forma:

- Modelo do Sistema - representa o modo de funcionamento do sistema, como as suas componentes estão integradas e como se conjugam para desempenhar o papel do sistema.

¹Human Computer Interaction

- Modelo do Designer - modelo que o *designer* da interface concebe e que pretende que seja transmitido pela interface. Este modelo deve ter em conta o utilizador final, em termos sociais e culturais.
- Modelo da interface - representa a forma como o sistema se apresenta ao utilizador.
- Modelo do utilizador - representa a forma como o utilizador imagina o funcionamento do sistema.

O *designer* pretende que o modelo do utilizador seja similar ao modelo do desenho. Contudo, a comunicação entre o designer e o utilizador não é directa sendo efectuada através do modelo da interface. Este deve tornar o modelo de desenho claro e consistente para o utilizador, evitando que este crie o modelo conceptual errado. É importante que o designer escolha o modelo conceptual apropriado e o comunique correctamente ao utilizador. Norman define alguns conceitos que deverão ser levados em consideração no desenho de interfaces:

- **Affordances**: propriedades dos objectos que permitem determinar a forma como devem ser utilizados. Os elementos devem ser auto-explicativos, ou seja, a sua aparência deve sugerir ao utilizador o que deve fazer, por exemplo um botão, que serve para premir, ou uma cadeira, que serve para sentar.
- **Mapping**: relação entre os elementos de controlo e o efeito que produzem no sistema.
- **Visibility**: que prevê que todos os elementos da interface estarão em locais perceptíveis e de fácil acesso.
- **Constraints**: que visam restringir as possibilidades de acção do utilizador, guiando-o para as opções correctas e diminuindo o espaço cognitivo.
- **Feedback**: as acções do utilizador devem produzir reacções imediatas e visíveis por parte do sistema, mantendo o utilizador informado do estado e dos resultados das suas acções.

Os vários tipos de *feedback*, resultado de uma acção efectuada por um utilizador, variam de sistema para sistema tendo em conta os seus objectivos. Assim, cada sistema interage de forma ímpar com o utilizador. Mesmo que inconscientemente, a própria interacção é desencadeada por uma série de eventos estimulados por *feedback*, ou seja, para cada acção existe um estímulo que pode proporcionar uma nova acção.

2.1.1 *Feedback* na Interação Humana

Ao interagir com um Humano, um sistema tem que possuir a capacidade de responder adequadamente às instruções dadas, independentemente do tipo de acção imposta pelo utilizador. Nos computadores actuais, o utilizador espera sempre que o sistema responda a qualquer acção desencadeada de forma rápida, para que o utilizador não se precipite e não execute novamente o mesmo pedido. É bastante importante que um *feedback* seja dado num momento oportuno, para que não haja complicações na interacção do utilizador com o sistema. Tomemos de exemplo o caso de um elevador. Ao pressionarmos o botão para o andar pretendido esperamos que o elevador execute a acção pretendida. O *feedback* é dado por um trancar da porta ou por um sinal sonoro, e até mesmo pelo impulso inicial, necessário para o arranque do elevador. Neste caso o utilizador irá saber que se está a movimentar, como pretendido. Ao chegar ao destino, é natural que o elevador apresente o andar onde se encontra ou, de uma maneira visual, apresentando o número do andar, ou de maneira sonora, dizendo o número do piso onde se encontra, obtendo-se assim a resposta final a uma acção desencadeada. Desta forma, é possível fazer com que o utilizador saiba exactamente o que se está a passar e qual o estado do seu pedido. O que acontece em muitos casos, é que os efeitos colaterais provocados por um tipo de *feedback* inadequado, podem ser traduzidos em condições não desejáveis. Temos o caso do consumo de energia eléctrica efectuado pelo consumidor, em que os valores apenas são verificados de dois em dois meses, podendo impedir o utilizador de alterar o seu comportamento, evitando o uso abusivo de energia atempadamente. Para evitar este acontecimento, Wood [29], apresenta os ECI² que, por meios de um *feedback* adequado, provam a melhoria comportamental de um utilizador. Assim, pode-se assumir que quando um utilizador recebe um estímulo desencadeado por uma acção sua, este estará disposto a levá-lo em consideração, alterando muitas vezes o seu comportamento. Comportamentos estes, como a poupança de energia, ou outros temas ambientais de grande importância, que poderão fazer parte do quotidiano, desde que o consumidor seja estimulado para tal, e que obtenha o *feedback* na altura correcta para desencadear tais acções. Quando o *feedback* é dado pelo sistema, no momento da acção, este denomina-se de *feedback just-in-time*, conceito que Arroyo [7] explora nos dois sistemas seguintes. O WaterBot é um dispositivo que, ligado a uma torneira, poderá dar indicações dos gastos de água numa casa e, inclusivamente, comparar com o gasto de vizinhos, de maneira imediata, permitindo ao utilizador ter uma reacção contígua. O sistema HeatSink (figura 2.2) altera a cor da água com

²energy-consumption indicators

efeitos luminosos, indicando ao utilizador a temperatura da água, sendo que a percepção visual do utilizador é imediata.

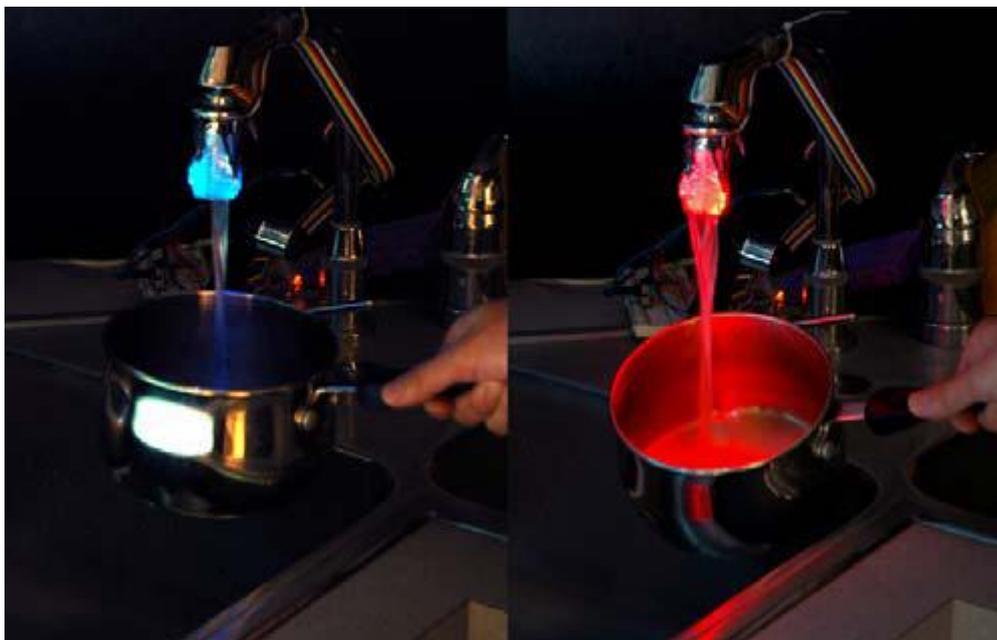


Figura 2.2 Demonstração de HeatSink, retirado de [7]

2.1.2 Tipos de *Feedback*

Desde sempre que se pretende manter o utilizador informado de maneiras mais apelativas e de forma mais cómoda, requerendo um menor espaço cognitivo. Tendo em conta as características físicas do ser Humano, é possível caracterizar alguns tipos de *feedback* tais como:

- **Visual:** que se baseia nas capacidades da visão do ser Humano. Esta é uma das formas de *feedback* mais utilizadas e a mais fácil de ser manipulada, sendo que normalmente o utilizador tem sempre uma percepção visual do sistema ou da interface do mesmo. Por exemplo, um semáforo de controlo de velocidade, que altera a sua cor de um amarelo intermitente para um vermelho contínuo, indica que o condutor está a transgredir a velocidade máxima no local.
- **Auditivo:** a par do visual, este tipo de *feedback* é bastante utilizado, em muitos casos, como um método auxiliar de *feedback*. Por exemplo, uma linha telefónica que possui

dois estados diferentes: a chamar ou ocupado. Desta forma, quando a linha se encontra a chamar, o sinal sonoro que o utilizador ouve é diferente do sinal de uma linha ocupada podendo, então, distinguir o estado em que a linha se encontra.

- Táctil: este tipo de *feedback*, é menos utilizado mas pode tornar-se essencial em diversas situações. Por exemplo, para os invisuais foi desenvolvido um sistema, o Braille³, que representa uma linguagem textual de reconhecimento através do tacto. Outra forma que poderá ser mapeado para este tipo de *feedback* é o vibrador de um telefone que, através do toque vibratório, permite alertar o utilizador da chegada de uma chamada. Krol [13] define ainda um conceito subjacente ao *feedback* táctil, que se traduz quando os sensores cinestésicos fazem a principal recepção do *feedback*. Ou seja, sempre que este seja percebido por factores físicos e não lógicos. Dando como exemplo o *force-feedback*⁴, que é muito utilizado em dispositivos de jogos como volantes, fazendo com que estes se voltem a centrar, isto é, todo o movimento é sentido pelos músculos do braço e antebraço.
- Combinação dos anteriores: este tipo corresponde à conjugação dos anteriores. Um exemplo desta conjugação é o telefone móvel que, ao receber uma chamada, poderá emitir um som (toque de chamada) acompanhado de um piscar do ecrã, representando o *feedback* visual, e a vibrar em simultâneo. Este tipo de *feedback*, permite que o utilizador possa saber que está a receber uma chamada independentemente da localização do telefone. Ou seja, se estiver a uma distância que não se possa ouvir ou em locais com bastante ruído de fundo, é possível verificar a luminosidade do ecrã ou, se não estiver a ver o telefone, ouvir a música ou sentir a vibração do mesmo.

³sistema de leitura com o tacto, para invisuais inventado pelo francês Louis Braille

⁴informação interpretada por partes corporais sensoriais de larga escala

2.2 Sistemas Ubíquos

Mark Weiser [28] descreve os sistemas ubíquos como tecnologias que se integram no ambiente quotidiano de um ser humano, tornando-se parte dele. O ser Humano tende a interiorizar, tornando usual, tudo o que consiga dominar. Desta forma, a invisibilidade do sistema é uma consequência fundamental da psicologia humana. Weiser [28] estabelece um novo paradigma de interacção pessoa-máquina, pelo que são os computadores que se integram no ambiente humano, e não o Humano que tem de aprender a lidar com os computadores, chamando-lhe **computação ubíqua**. Este tipo de sistemas poderá trazer grandes vantagens para os problemas que existem no dia-a-dia. Intille [9] descreve como se poderia adaptar a computação ubíqua à alteração do comportamento no cuidado da saúde. Para tal, descreve três tipos de sistemas que poderiam ser adaptados num sistema hospitalar. São eles:

- **Detectores de crises**, sendo necessário existirem vários sensores espalhados pelo corpo, como sensores de calor ou sensores biométricos, podendo ainda ser implementado a monitorização cardíaca para, assim, indicar a necessidade de assistência médica. Desta forma, este sistema estaria pronto a detectar uma alteração abrupta na condição física do utilizador indicando a necessidade de assistência médica imediata.
- **Detectores de agravamento do estado de saúde**, foi implementado um protótipo que monitoriza, de forma autónoma, a condição física de um paciente detectando gradualmente a existência de possíveis agravamentos.
- **Motivadores de comportamento saudável**, assim para além de medir os comportamentos de um paciente, o sistema poderá também intervir, alertando para eventuais problemas que poderão surgir, levando a que o paciente altere o seu comportamento.

Um sistema ubíquo, para conseguir manter-se de uma maneira translúcida no ambiente Humano, necessita tratar a abordagem na interacção pessoa-máquina com cautela para não se tornar num sistema intrusivo, algo que é do desagrado da sociedade em geral. Desta forma, Kranz et al. [12] desenvolveram um sistema, *AwareKitchen*, integrado numa cozinha, com o objectivo de auxiliar um utilizador nas tarefas a executar neste espaço, como visível na figura 2.3. A *AwareKitchen*, encontra-se equipada com vários sensores capazes de verificar acções do utilizador e detectar objectos na cozinha. Assim, o sistema permite, sem que o utilizador necessite de alterar o seu comportamento natural na utilização da cozinha, o reconhecimento de objectos que

estão a ser utilizados, bem como os alimentos ou a possibilidade de saber se existem portas dos armários abertas ou detectar se o fogão se encontra ligado. Este sistema é um exemplo muito concreto da definição, descrita acima, de um sistema ubíquo, visto que toda a interacção não obriga o utilizador a conhecer as idiossincrasias do sistema.



Figura 2.3 Exemplo de *AwareKitchen*, retirado de [12]

2.3 Sistemas Persuasivos

2.3.1 Objectivos

Segundo Fogg [5] a persuasão é uma tentativa de alterar atitudes, comportamentos ou ambos. Desta forma, é visto que um sistema persuasivo tem como função a alteração de comportamentos que um ser Humano possui de forma intrínseca, de uma maneira lúdica, consciencializando-o através da interacção com um sistema. A persuasão é utilizada muitas vezes, em campanhas publicitárias para promoção de produtos ou causas como por exemplo, na reciclagem ou protecção do meio ambiente, onde figuras públicas demonstram como se faz incentivando a fazer, ou porque é bom para o ambiente, ou porque não é difícil e/ou incómodo. Desta forma, as pessoas ficam sensibilizadas, aderindo. Fogg [5] cria o conceito de *Captology*, ("*Computers as persuasive technologies*") que foca o desenho, a investigação e a análise de sistemas computacionais interactivos criados com o objectivo de alterar atitudes e comportamentos humanos.

2.3.2 Métodos de persuasão

Existem várias formas de persuadir, dependendo do contexto sócio-cultural em que o alvo de persuasão está inserido. Segundo Harjumaa e Oinas-Kukkonen [8] existem três tipos de persuasão: *interpersonal*, *computer-mediated* e *human-computer*. Distinguem-se dois pólos de interacção: o persuasor, que tenta persuadir, e o persuadido, que é levado a alterar o seu comportamento. Persuasão *interpersonal* ocorre quando existe mais do que uma pessoa a interagir com outra. Desta forma, através de aspectos comportamentais e verbais o persuasor faz com que outra pessoa seja persuadido. Persuasão *computer-mediated*, ocorre quando o persuasor e o persuadido tenham como meio intermédio de comunicação um sistema computadorizado, como por exemplo, através de correio electrónico. Na persuasão *human-computer*, o persuasor é um sistema informático que tenta cativar o utilizador a alterar o seu comportamento. Embora seja necessário ter em consideração as limitações intrínsecas a estes sistemas, eles proporcionam grandes vantagens na persuasão em relação a um ser Humano, tais como:

- Serem mais persistentes, conseguindo repetir por vezes indeterminadas a mesma tarefa.
- Possuem várias formas de *feedback*, áudio, vídeo ou gráfico, que poderão influenciar o utilizador, não só na informação a receber, mas também na forma como esta é apresentada.

- Guardar e processar grandes quantidades de informação.
- Garantir o anonimato, sendo que o utilizador poderá interagir com um sistema sem revelar a sua identidade.
- Capacidade ubíqua, que lhes permite actuar em locais que não são apropriados para o ser humano, mas que são os locais correctos onde actuar como por exemplo, nas casas de banho ou em locais religiosos.
- Possuem escalabilidade, podendo ser replicados para vários pontos, mantendo o mesmo grau de interacção e persuasão.

As interligações das três componentes acima referidas podem ser facilmente compreendidas na figura seguinte.



Figura 2.4 Interligação de tipos de persuasão, adaptado de [8]

Para o conceito de *Captology*, Fogg [5] descreve sete tipos de métodos persuasivos, que permitem a um produto interactivo ser desenhado para alterar atitudes, comportamentos ou ambos, tornando o resultado desejado mais fácil de atingir. São elas:

- *Reduction*, que consiste na decomposição de actividades complexas em tarefas simples ou em menos tarefas: por exemplo na compra em lojas on-line, os meios de pagamento *one-click* minimizam os passos que o utilizador tem que dar até ao pedido de envio do produto.

- *Tunneling*, que levam o utilizador a seguir um determinado número de passos predefinidos. Este método é muito utilizado nas instalações de programas que são feitas passo-a-passo.
- *Tailoring*, que leva a separar os conceitos em necessidades ou interesses individuais. Ou seja, ao invés de apresentar uma informação genérica, selecciona o tipo de informação a ser apresentada para o tipo de utilizador que se encontra a interagir.
- *Suggestion*, que representa uma sugestão interventiva num local estratégico, surgindo num momento oportuno.
- *Self-Monitoring*, que permite a um utilizador monitorizar algumas características próprias alterando o seu comportamento ou atitudes de forma a atingir objectivos predeterminados. Por exemplo, um relógio que permite contar os batimentos cardíacos, levando a que o utilizador possa moderar a sua actividade.
- *Surveillance*, que possibilita a monitorização de um individuo numa acção específica, por exemplo a verificação de regras de higiene dos trabalhadores de uma empresa de modo a assegurar que estes as cumprem devidamente.
- *Conditioning*, ou seja um sistema que utiliza princípios compensatórios para a alteração de comportamentos: Por exemplo, a recompensa a um animal após a execução de um truque.

2.3.3 Exemplos de sistemas persuasivos

Em alguns sistemas a persuasão é, como diz o lema, "*a alma do negócio*". Desta forma, todo o sistema ou grande parte do mesmo é composto por elementos persuasivos, quer para ensinar o utilizador, quer para o incentivar a manter-se ligado ao sistema. De seguida, são apresentados três exemplos de sistemas persuasivos, que possuem objectivos finais distintos.

2.3.3.1 Facebook e Mixi

Estes dois sistemas têm por base persuadir o utilizador fazendo com que os seus espaços na Internet sejam permanentemente frequentados pelos mesmos. Desta forma Fogg e Iizawa [6]

estudam dois sites, o Facebook ⁵ mais utilizado nos Estados Unidos e o Mixi ⁶ mais popular no Japão, mostrando que estes se regem por quatro objectivos de persuasão em comum, sendo que em alguns casos a estratégia é diferente. São eles:

- **Criação de uma página de perfil**, tentando fazer com que o utilizador crie valores e conteúdos no site. Através de interfaces gráficas convencionais simples, ambos os sistemas tentam persuadir o utilizador a ser o mais específico possível no seu perfil, pois a qualidade e quantidade de informação de cada site encontra-se nessas mesmas adições.
- **Convidar amigos**, envolvendo mais utilizadores, contribuindo para o aumento da população de utilizadores do site. Em ambos os sites é possível convidar amigos a efectuarem um registo. O Facebook utiliza métodos mais agressivos que implicam a cedência, por parte do utilizador, das credenciais da sua caixa de correio electrónico para poder aceder à lista de contactos e enviar para todos os utilizadores dessa lista. Quanto ao Mixi, utiliza uma abordagem mais suave, pedindo somente ao utilizador que coloque todo o endereço de correio electrónico de um possível amigo.
- **Colocar conteúdo específico no site**, tentando fazer com que o utilizador crie valores e conteúdos nele, de uma forma específica para cada utilizador. Facebook e Mixi permitem a adição de ficheiros e/ou hiperligações, aumentando assim o interesse por parte de outros utilizadores, de comentarem os conteúdos adicionados.
- **Regresso ao site com frequência**, criando uma fidelidade, por parte do utilizador, com o site. Neste caso, são analisadas três formas de motivar os utilizadores a usarem o site frequentemente, são elas:

Persuasão via correio electrónico, em que os sistemas avisam o utilizador das alterações que estão a ser feitas, em termos de comentários dos amigos aos seus conteúdos ou de adição de conteúdos, por parte dos seus amigos, e que o utilizador poderá comentar.

Persuasão através do resumo das actividades dos amigos, neste caso, é proporcionado ao utilizador a visualização do que os seus amigos, que estejam ligados, estão a fazer ou têm agendado para o dia, como as alterações a fotos ou as componentes do perfil.

Persuasão através de *poked* e *ashiato*. No Facebook é possível efectuar um *poke*, acção que permite que um utilizador quando visita um perfil de alguém possa indicar, ou

⁵<http://www.facebook.com/>

⁶<http://mixi.jp>

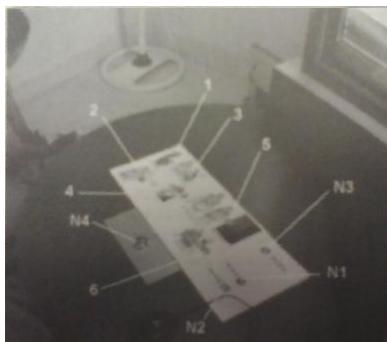


Figura 2.5 Exemplo de interface PHI, retirado de [20]

marcar, a sua presença como tal. De uma maneira semelhante, no Mixi o *ashiato*, visto como pegadas deixadas no site, indica a um utilizador que um outro visitou o seu perfil. Em ambos os casos, de uma maneira mais directa (*poked*) ou mais indirecta (*ashiato*), é possível persuadir o utilizador a frequentar um site, provocando a interacção entre os utilizadores e amigos.

2.3.3.2 *Personal Health Information*(PHI)

Parmar et al. [20] descrevem o uso de uma interface persuasiva para a ajuda do sistema nacional de saúde, no que toca a educação da população rural na Índia. O objectivo deste sistema, PHI, é educar as mulheres indianas, maioritariamente do mundo rural, alertando-as para as preocupações em tempo de gestação, assim como a explicação de todo o ciclo menstrual. Este sistema foi especialmente desenhado para conseguir alterar o comportamento das mulheres através de estratégias persuasivas intrínsecas ao sistema.

Na figura 2.5 é possível verificar uma parte integrante do sistema, a interface, desenvolvida para que a mulher, aquando da exploração do sistema, possa ter noção do que é pretendido e do assunto que irá ser abordado. Neste caso, a interface é dividida em dois tipos de botões: os funcionais, numerados de um a seis, e os de navegação, numerados pelos identificadores seguintes:

- **N1:** Botão que serve para navegar nas apresentações por diapositivos existentes no sistema.
- **N2:** Botão que serve para sair das apresentações por diapositivos existentes no sistema

- **N3:** Botão que tem o efeito do botão esquerdo de um rato, servindo para escolher o conteúdo pretendido.
- **N4:** Uma *trackball* para seleccionar hiperligações e fotografias.

Através dos botões funcionais é possível aceder a alguma informação que utiliza estratégias diferentes na persuasão do utilizador. Na tabela seguinte, é possível verificar essas estratégias mapeadas para cada botão.

N.º Botão	Informação	Estratégias de persuasão
1	Informação sobre o processo menstrual e saúde maternal	Pistas psicológicas; Linguagem; Feedback Audio-Visual
2	Informação sobre os problemas relacionados com o ciclo menstrual e saúde maternal	Análise gráfica descrevendo as piores condições de saúde; Pistas psicológicas; Linguagem; Feedback Audio-Visual
3	Boas práticas relacionadas com a menstruação e saúde maternal	Pistas psicológicas; Linguagem; Feedback Audio-Visual
4	Como as mulheres em países desenvolvidos lidam com problemas idênticos	Pistas psicológicas; Linguagem; Feedback Audio-Visual
5	Mostragem de filmes relacionados	Pistas psicológicas; Linguagem; Feedback Audio-Visual; Utilização de musicas tradicionais
6	Vídeo de líderes e profissionais de medicina para motivar as mulheres	Feedback Audio-Visual

Tabela 2.1 Estratégias de persuasão de PHI, adaptado de [20]

Da tabela anterior podemos extrair os seguintes conceitos de estratégias de persuasão, utilizados pelo PHI:

- **Pistas psicológicas:** nos vídeos demonstrativos e educativos são utilizados personagens adequadas para explicar os temas a serem abordados; por exemplo, uma adolescente para abordar o tema da menstruação ou um médico para responder a perguntas mais frequentes.
- **Linguagem:** tendo o PHI a capacidade de produzir *feedback* audiovisual, é necessário em alguns temas utilizar uma linguagem conveniente, tendo em conta a classe social do utilizador. Desta forma, no final de cada explicação, é possível incentivar as mulheres a seleccionarem outro botão para continuarem a sua formação através do PHI.

2.3.3.3 *Calorie-aware Kitchen*

Cada vez mais a alimentação é uma fonte de preocupação para a saúde. Assim, as pessoas tendem a avaliar a sua refeição tendo em conta as calorias que esta possui. De modo a facilitar o cálculo das calorias e no sentido de ajudar as pessoas a prepararem uma refeição caloricamente regrada, Pei-Yu Chi et al. [3] apresentam uma interface que possibilita o cálculo do número de calorias no momento da preparação de uma refeição. No sistema apresentado para a *Calorie-aware Kitchen* é utilizada uma bancada com sensores de peso, através dos quais é possível determinar o peso do alimento a adicionar à refeição, visível na figura 2.6. Como meio de melhorar a certeza do peso, eliminando o ruído produzido pelo corte do alimento ou outra actividade, é utilizada uma câmara para análise vídeo das actividades a decorrer no momento. Para que o utilizador possa ter o *feedback* da quantidade de calorias de cada ingrediente e da refeição até ao momento, existe um ecrã na bancada que vai mostrando esses valores actualizados a cada dois segundos.

Através desta interface os utilizadores conseguem controlar de uma forma natural o valor calórico das suas refeições. O facto mais evidente de persuasão é que ao ter um *feedback* imediato do número de calorias presentes na refeição, o utilizador é persuadido a manter o valor nutricional aquando a adição de alimentos. No contexto de efeitos persuasivos, os utilizadores tentam preparar a refeição com base nas calorias que estão envolvidas, conseguindo então uma alimentação mais saudável. Num teste efectuado a alguns utilizadores foi possível verificar que com esta interface os utilizadores eram persuadidos a elaborarem refeições menos calóricas. Neste teste foram concebidos três menus. Cada um possui um valor nutritivo estimado previamente, ao qual poderia ser, posteriormente, comparado o valor nutricional de cada refeição. Assim, foram preparadas seis refeições, duas por menu, uma utilizando o sistema, outra não. Foi possível concluir que em todos os casos a percentagem de calorias em excesso foi muito menor quando se utiliza este sistema persuasivo.



Figura 2.6 Sistema de Calorie-aware Kitchen, retirado de [3]

2.4 Interfaces Tangíveis

Todos os sistemas computadorizados que envolvem interacção com os utilizadores necessitam de uma interface que sirva de intermediário. Essas interfaces tiveram uma evolução ao longo da história sempre com base nas necessidades do utilizador, começando pelas linhas de comandos, passando pelas interfaces gráficas e mais recentemente recorrendo ainda a novas tecnologias. Por exemplo, as interfaces tangíveis que permitem o controlo de elementos digitais através da manipulação de objectos físicos.

Ishii e Ullmer [10] assumem que as interfaces tangíveis irão contribuir para o aumento do mundo físico através da associação de informação digital aos objectos do quotidiano ⁷. Os autores estendem esta definição assumindo ainda que um sistema tangível não deverá diferenciar o dispositivo de entrada (*input*) e de saída (*output*), embora assumam que poderá haver restrições à definição em prol de um melhoramento da interacção. Um exemplo apresentado pelos autores é o ábaco⁸ a que se pode alterar o estado e que, a própria alteração, auto-demonstrativa da acção feita pelo utilizador.

Ullmer e Ishii [27], descrevem a arquitectura de uma interface tangível (TUI) comparando-a com as interfaces gráficas (GUI). Nas figuras 2.8 e 2.7 podem ver-se os esquemas das arquitecturas dos modelos de ambas as interfaces. Ambas as arquitecturas têm em consideração o paradigma do espaço físico e digital, mas de maneiras diferentes. Todo o conteúdo, em termos de informação e processamento, é designado por modelo e, em ambos os casos, encontra-se em meio digital. A manipulação das interfaces é mapeada na arquitectura pelo controlo, que é notória a diferença em ambos os casos. Enquanto nas GUI o controlo se faz através dos dois paradigmas, físico e digital, nas interfaces tangíveis somente se necessita de interagir no físico. Por fim, a arquitectura clarifica o modo de visualização do resultado das acções efectuadas no sistema, em que para as GUI é representado pela visão e nas TUI repartido pela representação física e digital. Estas representações, física e digital, são distinguidas nas TUI de forma a que a representação física simboliza a orientação e o estado do objecto, enquanto a digital representa o conteúdo a ser visualizado no próprio objecto.

Nas interfaces mais usuais dos dias de hoje, as GUI, existem alguns elementos que foram criados para permitir uma interacção, mais adequada a cada caso, do utilizador com a dita

⁷"augment the real physical world by coupling digital information to everyday physical objects and environments"

⁸Aparelho para efectuar cálculos, utilizado como um calculador

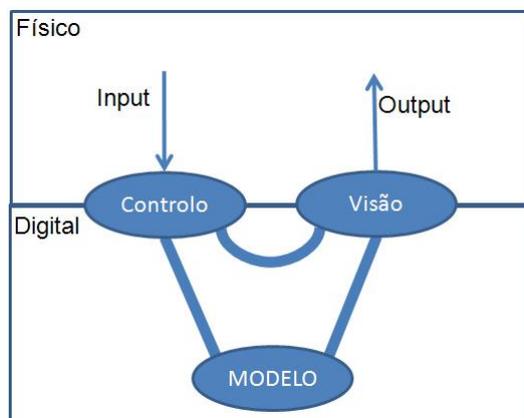


Figura 2.7 Arquitectura de uma GUI

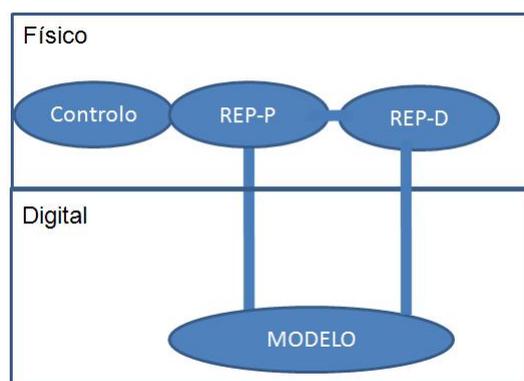


Figura 2.8 Arquitectura de uma TUI

interface. É natural em ambiente gráfico visualizar em janelas, com os respectivos conteúdos, ícones, menus, *handle*⁹ e barras de deslocamento para se visualizar o conteúdo que ocupe mais do que o tamanho da janela. Para todos estes elementos, Ishii descreve em [10] o mapeamento deste elementos representativos de GUI para elementos, objectos, nas TUI (figura 2.9). Assim, pode-se verificar que as acções de cada elemento das GUI, e respectivo mapeamento nas TUI, são idênticas. No entanto, a diferença na interacção encontra-se na forma de manipular esses elementos, bem como o *feedback* dado ao utilizador.

No seguimento do estudo, feito por Maher e Kim [16], foram colocados dois designers a interagir com interfaces TUI e GUI. Assim foi possível verificar que a interacção com as GUI apenas é feita por um dos elementos, estando o outro apenas a apoiar, mas sem nunca intervir

⁹elemento que serve para seleccionar conteúdos em janelas e num ambiente de trabalho, ver figura 2.9

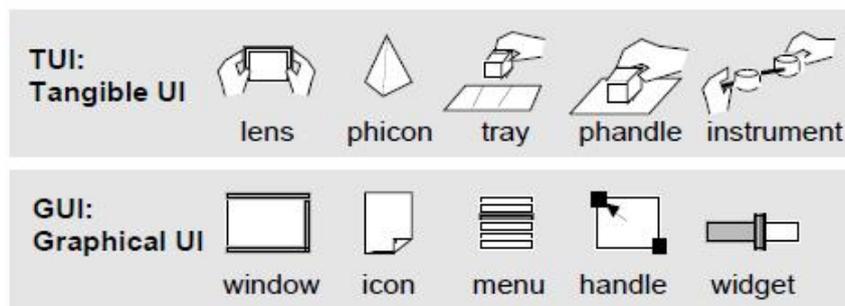


Figura 2.9 Mapeamento de GUI para elementos TUI, retirado de [10]

em alguma alteração na interface. Aquando da exposição a uma aplicação com TUI ambos os utilizadores conseguiram manipular os objectos simultaneamente. Sendo esta, por si só, uma vantagem para as TUI. Contudo, com as TUI não é permitido criar objectos de interacção dinamicamente, facto que é possibilitado aquando do uso das GUI. Por exemplo, nas GUI é permitido o aparecimento de ícones após uma acção. Este facto não é possível nas interfaces TUI, a menos que sejam fisicamente colocados os *phicons* pelo utilizador. Ao utilizar sistemas TUI para a visualização e manipulação de objectos 3D, foram detectados alguns comportamentos típicos que permitem verificar que estas interfaces se mostram, em certos casos, mais eficazes do que as GUI. Estes comportamentos passam pela execução das acções num menor espaço de tempo e maior produção de acções de interpretação.

2.4.1 Exemplos de Sistemas Tangíveis

2.4.1.1 Tangible Newspaper

Sporka et al. [24], apresentam o Tangible Newspaper. Trata-se de um sistema tangível concebido para utilizadores com incapacidades visuais. Este sistema baseia-se na forma como algumas crianças, em fase de aprendizagem, usam os dedos indicadores para facilitar a orientação da leitura. Neste sentido, o utilizador possui uma etiqueta de *Data Matrix*¹⁰ no dedo indicador que permite ao sistema verificar qual a zona do jornal que é necessária reproduzir. Todas as zonas do jornal encontram-se em memória, de forma a melhorar a eficiência do sistema. Então, através da síntese de TTS¹¹ é possível apresentar ao utilizador, via áudio, o conteúdo que se

¹⁰http://en.wikipedia.org/wiki/Datamatrix_code

¹¹text-to-speech, ou passagem de texto para áudio

apresenta na zona do jornal seleccionada.

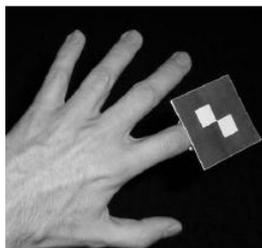


Figura 2.10 Exemplo de *Data Matrix* para o Jornal Tangível, retirado de [24]

Desta forma, toda a arquitectura envolvente ao sistema é constituída, para além do indicador, composto por um código *Data Matrix* associado ao dedo indicador, mostrado na figura 2.10, por uma câmara. Esta Câmara é utilizada para duas funcionalidades: uma para detectar a posição do indicador, outra para verificar o tipo de formatação que a página do jornal apresenta, possibilitando o mapeamento da posição do indicador, para uma área do jornal, identificando-se o texto que o utilizador pretende que seja reproduzido. Uma dificuldade que se encontra reside na luminosidade que deve ser sempre a ideal, pois a detecção dos padrões de *Data Matrix* depende dessa componente. Este sistema possui alguns formatos de páginas de jornal, predefinidos em memória, para que possa haver mapeamento do texto de uma forma mais eficaz, sendo desenvolvido com o ARToolKit [2], para a verificação e determinação da posição do indicador, do utilizador e, conseqüentemente, da zona do jornal a reproduzir.

2.4.1.2 Cube

Os jogos são um dos métodos mais utilizados para a aprendizagem de uma criança. Assim sendo, Terrenghi et al. [26] apresentam uma interface tangível que permite a aprendizagem através de um jogo, sendo esta interface um simples cubo. O cubo por si só é um dos objectos mais fáceis de manipular e de entender por qualquer pessoa, o que potencia a capacidade deste ser implementado como interface tangível. O cubo, como é apresentado na figura 2.11, possui um ecrã em cada face, um altifalante no seu interior e, ainda, sensores de aceleração que permitem verificar se o cubo está a ser agitado e qual a face voltada para cima. Ora, lançando um desafio ao utilizador, por exemplo uma pergunta num dos ecrãs, este poderá responder colocando a resposta correcta que estará apresentada numa das restantes faces, virada para cima, e

abanando o cubo vigorosamente para validar a pergunta. Ao ser detectado o movimento, o sistema verifica se é ou não a resposta correcta emitindo o *feedback* correspondente. Esta interface permite ainda definir as respostas e perguntas de uma forma textual ou através de imagens. Esta interface apresenta problemas quanto à forma de validação (abanar o cubo), pois os utilizadores tinham alguma dificuldade em perceber este método optando, primordialmente, por gesticular em frente à resposta correcta.

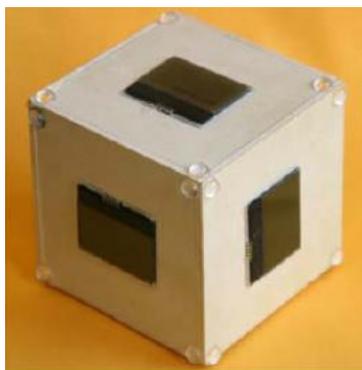


Figura 2.11 Exemplo de cubo como interface Tangível, retirado de [24]

2.4.1.3 *Sensetable*

Muitos sistemas de interfaces tangíveis utilizam câmaras para a localização de objectos. Estas aproximações têm dificuldades em ambientes com luminosidade variada ou extrema (muito escuro, muito claro) onde a câmara, não estando preparada para a alteração de luminosidade, não consegue detectar os objectos. Dificuldades essas que também são proporcionadas quando os objectos são movimentados pelo espaço, onde a latência da localização, do sistema, é maior, podendo afectar o desempenho da aplicação. Para contornar estes tipos de problemas Patten et al. [21] desenvolveram uma plataforma denominada *Sensetable* que possui interfaces tangíveis como meio de interacção, para que os utilizadores possam manipular a aplicação em grupos e com ambas as mãos, como pode ser verificado na figura 2.12.

Esta plataforma adquire uma arquitectura muito semelhante a uma mesa. Possui uma superfície onde é projectado o ambiente a ser manipulado pelo utilizador que através dos objectos, possui também dois monitores para o auxílio à aplicação, onde poderão ser apresentadas funcionalidades complementares (figura 2.13). Como a localização é feita através de sensores que se encontram na superfície da *Sensetable*, para além de esta ter uma latência muito menor possui

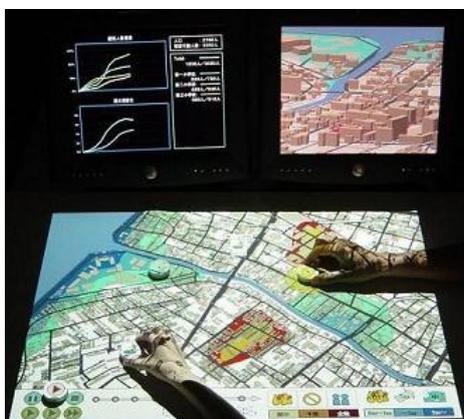


Figura 2.12 Exemplo de *Sensetable*, retirado de [21]

também a capacidade de indicar com precisão o posicionamento do objecto. O uso de objectos que são conectados sem fios, permite uma comunicação e manipulação sem interferências de cabos, melhorando por sua vez a interação que o utilizador tem com as aplicações.

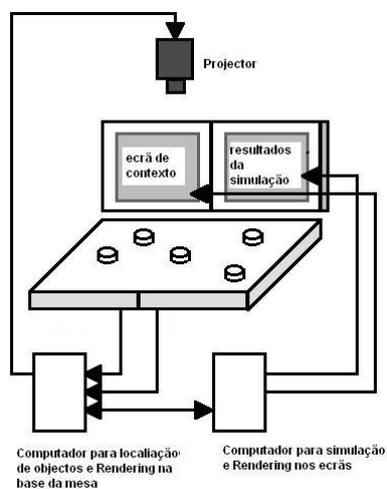


Figura 2.13 Arquitectura de *Sensetable*, retirado de [21]

2.4.1.4 *Reactable*

Jordà et al. [11] apresentam a *Reactable*, um novo instrumento musical que utiliza objectos tangíveis para conseguir produzir os sons desejados. A *Reactable* permite uma utilização por

múltiplos utilizadores e a manipulação de vários objectos em simultâneo. Cada objecto funciona como um sintonizador, possui uma funcionalidade diferente na aplicação, podendo gerar, modificar ou controlar o som. Como pode ser verificado na figura 2.14, a *Reactable* funciona como uma mesa, que interage com o utilizador através dos objectos descritos acima. Pode também ser visto que por baixo da mesa existem dois componentes importantes para a arquitectura desta interface. São eles a câmara, que tem como objectivo a verificação da posição dos objectos, e o projector que projecta as imagens de *feedback* para a mesa. Por outro lado, existe sempre um altifalante que emite o som, dependendo da disposição dos objectos na mesa.

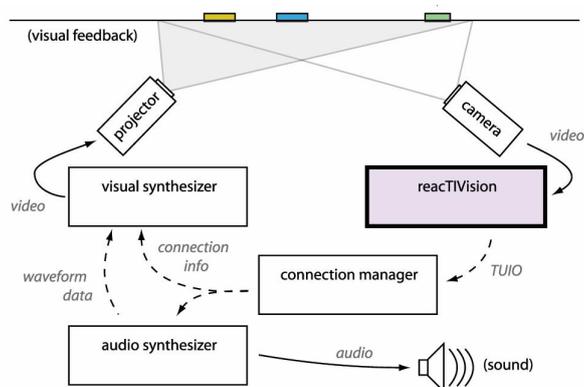


Figura 2.14 Arquitectura de *Reactable*, retirado de [11]

Cada objecto pode conectar-se ou desconectar-se a outro através da proximidade para com este, assim é possível associar um objecto a outro. Cada objecto possui uma aura em seu redor que serve para informar o utilizador do estado do objecto, permitindo ainda alterar esse mesmo estado, como por exemplo, aumentar ou diminuir o volume sonoro. A *Reactable* apresenta algumas vantagens devido à sua arquitectura e idiossincrasias do seu sistema, como por exemplo:

- Possibilidade de um sistema compacto, onde a câmara e o projector se encontram escondidos.
- Um número elevado de objectos a serem adicionados, sendo o tamanho da mesa a limitar esse número.
- Possibilidade de detectar a orientação dos objectos, não sendo tratados como meros pontos.

- Adição simplificada de novos objectos, e detecção de dedos, para a manipulação do estado de cada objecto.

São já comercializados sistemas tangíveis que se afirmaram como uma boa alternativa aos métodos convencionais de interacção com o utilizador. Por exemplo, a *Microsoft* desenvolveu a *Surface* ¹², uma mesa que interage com o utilizador através de toque, tendo a possibilidade de detectar vários pontos de toque e permitindo a interacção com mais do que um utilizador em simultâneo. Desta forma, permite aplicações como jogos de casino ou venda de produtos que tiram partido das vantagens deste sistema.

¹²<http://www.microsoft.com/surface/>

2.5 Interação com crianças

Cada vez mais as crianças são rodeadas de sistemas computacionais desenvolvendo, assim, mais cedo as capacidades necessárias para lidar com estes. Contudo, a percepção da criança para com os objectos envolvidos nos sistemas computacionais difere da percepção de um adulto. Assim, quando se desenvolve algo que esteja focalizado para crianças, é necessário ter em conta as características físicas e psicológicas desta faixa etária, para que todo o produto possa ser concebido com base na possibilidade de uma boa interação e percepção por parte da criança. Segundo Read [18], a interação com crianças, denominada de CCI ¹³, é um ramo da HCI, apresentado na secção 2.1, que tem como objectivo estudar a forma como as crianças utilizam os produtos interactivos. Para Read [22], a avaliação da interação é importante para determinar e assegurar a usabilidade de um produto. Assim, Read apresenta algumas razões pelas quais se deve envolver as crianças no desenvolvimento de aplicações interactivas a elas destinadas, tais como:

- A diferença da percepção do mundo por parte das crianças e dos adultos, podendo estes últimos não perceber o que as crianças pretendem de um produto.
- As crianças são actores e participantes activos da sociedade e não meros espectadores pelo que, cada vez, mais se incluem as crianças no processo de decisão que leva à construção dos seus ambientes computacionais.
- O envolvimento no desenho e avaliação de um produto tem-se revelado uma experiência interessante, motivadora e recompensadora quer para os investigadores, quer para as crianças.

Para que a avaliação da interação de crianças seja feita, de uma forma concreta e concisa, é necessário ter em conta determinados aspectos. Normalmente, neste tipo de avaliações, para além da avaliação aquando da interação em termos de uso, como a linguagem corporal e as dificuldades apresentadas durante a interação, o uso de questionários é bastante utilizado. Um número de perguntas reduzido, a certeza de que estas poderão ser respondidas e que a resposta será sincera são aspectos que têm necessariamente de ser levados em conta. Ainda assim, torna-se importante que a escala de avaliação seja apropriada. Desta forma, Read [22] apresenta o *Fun Toolkit*, uma ferramenta para ser utilizada na preparação de questionários efectuados

¹³Child Computer Interaction

especificamente para crianças. Assim, são apresentados três tipos de instrumentos que poderão ser abordados nos testes de avaliação à satisfação de uma criança por um produto ou actividade desenvolvida.

1. **Smileyometer:** Como pode ser verificado na figura 2.15, através deste instrumento, a criança é incentivada a escolher uma das caras, correspondendo ao seu estado de espírito. Esta escala poderá ser utilizada para responder a questões colocadas antes ou depois da interacção. Caso seja utilizada antes, poderá servir para medir o interesse que a criança tem em interagir com o produto. Se utilizada depois servirá para medir a satisfação da criança face ao produto.



Figura 2.15 Exemplo de *Smileyometer*, retirado de [22]

2. **Fun Sorter:** Consiste numa tabela em que o número de colunas representa o número de itens a serem comparados e as linhas as construções a serem comparadas pelo utilizador. Permitindo assim a avaliação não só da satisfação da criança, mas também a ordem de preferência por um produto. Assim, este instrumento é utilizado para comparar conjuntos de tecnologias ou produtos. Este instrumento oferece, ainda, a possibilidade de entreter a criança, através da ordenação dos itens enquanto responde ao questionário (figura 2.16).

	Best			Worst
Worked the best	writing	typing	speaking	skating
Liked the most	writing	typing	speaking	skating

Figura 2.16 Exemplo de *Fun Sorter*, retirado de [22]

3. **Again Again Table:** A figura 2.17 ilustra uma tabela que consegue indicar a vontade de a criança repetir uma dada acção, através da escolha de uma das possibilidades *sim*, *talvez* ou *não*. Desta forma, não se pode avaliar um só produto, mas comparar vários produtos em termos de satisfação e conseqüente usabilidade.

Would you like to do it again?

	Yes	Maybe	No
	✓		
		✓	

Figura 2.17 Exemplo de *Again Again table*, retirado de [22]

Fisch [4] descreve o *feedback* nos jogos lúdicos, como um meio importante para a aprendizagem de uma criança, quando esta tenta resolver um problema ou um jogo em ambientes que não lhe são familiares. Assim, na criação de um jogo é importante que o *feedback* dado, quando uma criança efectua uma acção errada, seja levado em consideração. Desta forma, o *feedback* não deverá ser somente a revelação imediata da acção correcta que deveria ter sido executada. Devendo esta ser apenas revelada caso as acções incorrectas se repitam por três ou quatro vezes consecutivas. Para que tal não aconteça, o *feedback* deverá ser formulado de forma a dar algum incentivo que guie a criança no sentido da resolução do problema. Tal terá um impacto positivo na criança mostrando o porquê da resposta estar errada ou ajudando a encaminhá-la para a solução do problema. De uma forma persuasiva, o *feedback* poderá encaminhar a criança a chegar ao objectivo pelos seus próprios meios. Fisch [4] descreve um exemplo de um jogo de imitação em que a criança é incentivada a repetir uma dada sequência de acções. Caso a sequência de acções não seja executada de forma correcta o *feedback* será em forma de incentivo, mostrando o que faltou na sequência. Mesmo quando não existe interacção por parte da criança, o sistema tenta incentivá-la a completar as sequências.

Recentemente, foi elaborado um estudo que compara as GUI, as TUI e o método tradicional, no que concerne à facilidade que as crianças têm em resolver problemas, neste caso um puzzle [1]. De notar que o desenho do puzzle de cada interface é exactamente o mesmo. Para as diferenças entre as características de implementação de cada interface pode observar-se a tabela 2.2. Nesta investigação foram utilizadas crianças, aos pares, para resolver os puzzles. Sendo que cada par de crianças poderia resolver o puzzle o número de vezes que quisesse, dentro de um espaço de tempo de quinze minutos. As crianças eram submetidas a um só tipo de interface, para os solucionar. O objectivo principal deste estudo centrava-se na demonstração de cinco hipóteses que permitiam comparar os três tipos de interfaces. Do referido estudo pode-se concluir o seguinte:

- Os pares que utilizaram as TUI e os que utilizaram o método tradicional resolveram pela primeira vez o puzzle, mais rapidamente.
- Existência de um maior número de utilizadores, que utilizaram as TUI, a terminar o puzzle pelo menos uma vez, comparativamente com as GUI. Embora se tenha verificado que terminaram mais vezes os utilizadores que resolviam o puzzle recorrendo ao método tradicional.
- Os utilizadores que resolveram os puzzles através das TUI passavam mais tempo a manipular as peças, estando mais tempo focados na resolução do puzzle, do que nas GUI, onde os utilizadores passaram grande parte do tempo a falar ou a gesticular.

	Tradicional	GUI	TUI
Manipulação directa do objecto	+	-	+
Multi-utilizador	+	-	+
Feedback audiovisual	-	+	+
Feedback tátil	+	-	+

Tabela 2.2 Diferenças nas características de implementação, adaptado de [1]

2.6 Modelo Predador - Presa

Um ecossistema é constituído pelos organismos vivos que habitam numa determinada área, o seu ambiente envolvente e as interacções que estabelecem entre si. Entre essas interacções contam-se as cadeias alimentares. Uma cadeia alimentar é uma sequência de seres vivos que se alimentam uns dos outros. Cada espécie ocupa um determinado nível numa cadeia alimentar e a função que desempenha nessa cadeia pode classificar-se em produtor (ex. plantas), consumidores (ex. animais) ou decompositores (ex. fungos ou bactérias). Entre os animais podem estabelecer-se relações predador-presa. Entenda-se por presas, os seres vivos que possam servir de alimento a outros seres vivos e por predadores aqueles que, por sua vez, se alimentam das presas. Segundo Rafikov [17], o sistema de Lotka-Volterra traduz um modelo descritivo das interacções predador-presa. Este modelo poderá ser aprofundado para interacções com múltiplos predadores e/ou múltiplas presas. Neste estudo vamos centrar-nos no modelo de Lotka-Volterra com um predador e uma presa, assim como é descrito por Takeuchi [25]. Desta forma, é descrito que o modelo de Lotka-Volterra é dado por duas equações diferenciais, que identificam o número de indivíduos ao longo das interacções. Equações essas que podem ser verificadas através da figura 2.18.

$$\frac{dx}{dt} = x(\alpha - \beta y)$$

$$\frac{dy}{dt} = -y(\gamma - \delta x)$$

X – Number of predators
 Y – Number of preys
 α, β, γ, δ – parameters of predator – prey interaction

Figura 2.18 Equações representativas do modelo de Lotka-Volterra

Sendo os parâmetros da figura constantes ao longo de todas as interacções, é possível verificar que o equilíbrio é mantido até ao infinito, caso não hajam alterações bruscas nas populações dos predadores ou das presas, constatável através do gráfico existente na figura 2.19. Figura esta que representa um simulador, em Excel, das interacções predador-presa. Este simulador permite, através da introdução de dados, na área realçada a vermelho, a alteração dos valores das populações iniciais e dos parâmetros das interacções predador-presa. A área a azul, apresenta

os valores de cada população após cada iteração do algoritmo, estes encontram-se representados no gráfico. Assim para cada iteração é avaliado um encontro entre predador-presa e a susceptibilidade que cada um tem, respectivamente para se alimentar, ou servir de alimento.

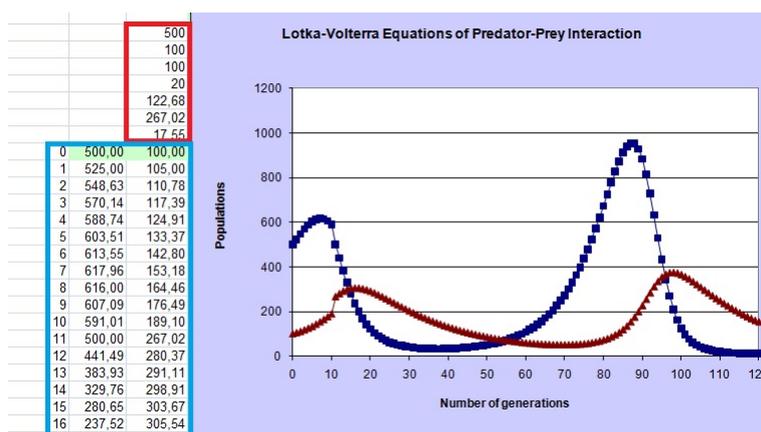


Figura 2.19 Excerto do ficheiro Excel que simula as interacções predador - presa, segundo o modelo de Lotka-Volterra

3 . Metodologia e Desenvolvimento da *Framework*

Este trabalho pretende estender o desenvolvimento da *Framework* para a criação de sistemas persuasivos, desenvolvida no âmbito da tese de mestrados de Pedro Lobo[15], intitulada de versão 1.0, de forma a superar algumas das suas limitações, nomeadamente permitir a construção de sistemas com interfaces tangíveis e a adição de sensores. Consideramos como sensores qualquer aplicação, algoritmo ou mecanismo que possam ser adicionados à *Framework*, para mais tarde serem utilizados pelas aplicações que esta executa. A *Framework* integra também uma *Authoring Tool* que foi alterada com o intuito de facilitar a visualização do grafo do sistema que o utilizador pretenda criar, permitindo a elaboração de sistemas de uma forma mais simplificada.

Assim, a metodologia seguida durante os trabalhos desenvolvidos nesta tese inclui os seguintes passos:

- **Revisão de literatura e trabalho relacionado**, neste passo foram analisados trabalhos anteriormente elaborados e foram revistos aspectos importantes a considerar ao longo da realização desta tese.
- **Estudo da *Framework* inicial**, onde foram verificados os aspectos técnicos que compunham o sistema.
- **Análise das alterações a efectuar**, neste passo foram analisadas as limitações que a *Framework* possuía, assim como as alterações que poderiam ser feitas à mesma.
- **Desenvolvimentos da nova *Framework***, neste passo foram implementadas as alterações à *Framework*, analisadas no passo anterior.
- **Desenho e desenvolvimento do caso de estudo**, de forma a validar todo o trabalho desenvolveu-se como caso de estudo um sistema (*Ecosystem Room*), descrito com maior pormenor na secção 5.1 deste documento.
- **Testes de avaliação**, após o desenvolvimento do *Ecosystem Room* foram feitos alguns testes com crianças, para verificação da usabilidade das aplicações geradas pela nova *Framework* e para entendimento das potencialidades persuasivas e de aprendizagem que as interfaces tangíveis possuem face às interfaces gráficas convencionais.

3.1 Arquitectura da *Framework*

Na versão 1.0 da *Framework* é apresentada uma arquitectura, figura 3.1 de quatro camadas: *Multimedia Display*, *Context Processor*, *Data* e *Sensors*. Como pode ser observado na figura 3.2, a arquitectura desta nova *Framework* mantém alguns dos conceitos da versão anterior, sendo que as camadas *Sensors* e *Context Processor* sofrem algumas alterações. Estas devem-se ao facto de ter sido introduzida uma nova camada, *Communicators*, explicada mais adiante. Desta forma as alterações foram:

- ***Context Processor***

- Passagem dos processadores dos sensores e da câmara (*Sensors Processor* e *Camera Processor*), para um único módulo *Sensor Pool*.

- Possibilidade do módulo *Context Processor* actuar sobre as camadas inferiores.

- ***Sensors***

- Passagem de cada sensor físico (*IR Sensors* e *Camera*) para um módulo representativo de todos os sensores (*Sensors*). Esta alteração ocorre devido ao facto de a *Framework* interpretar cada sensor físico, algoritmo ou mecanismo de uma forma genérica.

- Obrigação de um sensor físico possuir meios de comunicação com a nova camada implementada nesta versão, descrita mais adiante.

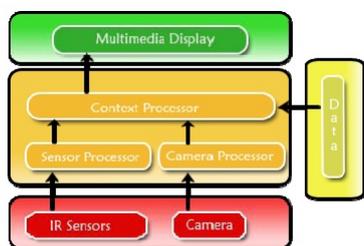


Figura 3.1 Arquitectura da *Framework* 1.0

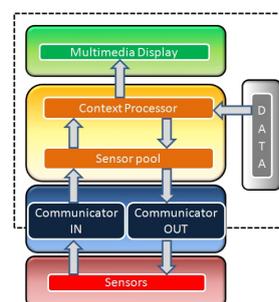


Figura 3.2 Arquitectura da nova *Framework*

Para um melhor entendimento de toda a arquitectura apresentada na figura 3.2, segue-se a descrição mais pormenorizada de cada uma das suas camadas.

- ***Multimedia Display***, responsável por apresentar ao utilizador o conteúdo multimédia (áudio e vídeo) da aplicação criada, sendo controlada pelo *Context Processor* que faz a avaliação contextual do ambiente envolvente.
- ***Data***, trata-se de uma camada que guarda toda a informação necessária para o bom funcionamento do sistema. Estes dados são tratados no módulo *Context Processor*.
- ***Context Processor***, esta é a camada mais importante da arquitectura, uma vez que toda a lógica e poder de decisão estão aqui implementados. Esta camada interpreta os dados dos sensores para, de seguida, tomar uma acção, tal como enviar informação para a camada superior ou para as camadas inferiores.
- ***Communicators***, esta nova camada introduzida possui, como se verifica na figura 3.2, dois módulos internos, *Communicator IN* e *Communicator OUT*, que podem ser vistos como um meio de comunicação, pois não se encontram explicitamente inseridos no código da *Framework*. A colocação desta camada, na explicação da arquitectura, serve para tornar visível a noção de comunicação entre os sensores e o núcleo da *Framework*. É através desta camada que, de forma padronizada descrita na secção 4.1, é permitido à *Framework* receber os dados dos sensores e/ou enviar informação para os mesmos, conferindo-lhe a possibilidade de ser genérica em termos de adição de qualquer sensor.
- ***Sensors***, esta camada representa os sensores em geral, como já foi enunciado, um sensor é entendido pela *Framework* como algo que lhe fornece dados contextuais independentemente da sua função intrínseca. Desde que possua a capacidade de comunicar com a *Framework*, através da camada superior (*Communicators*), qualquer sensor físico, algoritmo ou mecanismo poderá ser interpretado por esta como um sensor, permitindo enviar ou receber dados.

3.2 Arquitectura da *Authoring Tool*

O desenvolvimento de uma *Authoring Tool*, na primeira versão, foi vista como uma forma de facilitar a criação de aplicações por utilizadores com conhecimentos de programação limitados. Assim a *Authoring Tool*, criava ficheiros XML para que, mais tarde, estes fossem lidos em

tempo de execução para a tomada de decisão do *Context Processor*, que controlava o *Multimedia Display*. Nesta nova versão manteve-se a mesma ideologia de funcionamento da *Authoring Tool*, embora todo o aspecto gráfico, forma de edição e arquitectura do sistema tenham sido completamente alterados.

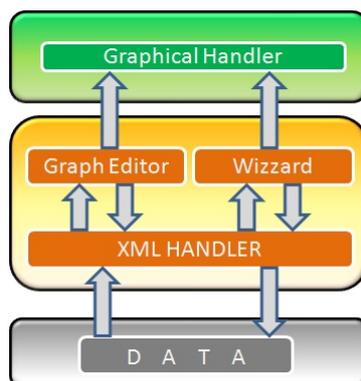


Figura 3.3 Arquitectura da *Authoring Tool*

Na figura 3.3, é possível diferenciar as três camadas principais da arquitectura da *Authoring Tool*. Segue-se a explicação detalhada para cada uma delas.

- **Graphical Handler**, esta camada é a responsável por todo o suporte gráfico que a *Authoring Tool* possui. É neste módulo que os formulários de preenchimento, para a configuração das aplicações a serem criadas, são alocados e apresentados ao utilizador. É também nesta camada que se faz a análise e respectivo desenho, tanto das ligações existentes no grafo como dos próprios estados.
- **Editor**, esta é a camada principal da *Authoring Tool* e é aqui que todo o processamento e análise são efectuados. Desta forma, a mesma é composta por três módulos. **Graph Editor**, que é o responsável pela análise dos grafos e por oferecer ao utilizador a possibilidade de os construir consoante as suas necessidades. **Wizard**, é neste módulo que se verifica quais os passos a tomar, nas janelas de configuração dos estados e onde se selecciona a informação a carregar para cada campo de preenchimento dos vários formulários. **XML Handler**, módulo responsável pela leitura e escrita de informação na camada inferior.
- **Data**, por fim este módulo representa todo o sistema de ficheiros montado para que a informação seja coerente e diferenciada de aplicação para aplicação. É neste módulo que

se encontra guardada toda a informação que o utilizador configura na criação da uma nova aplicação.

3.3 Diagrama de classes

Para um melhor entendimento da implementação dos sistemas, nesta secção são apresentados os diagramas de classes dos módulos mais importantes, tanto da *Framework* como da *Authoring Tool*.

3.3.1 Framework



Figura 3.4 Diagrama de classes da Framework

Na figura 3.4, são visíveis oito classes distintas que se assumem como sendo as mais importantes para o bom funcionamento da *Framework*. De seguida, procede-se a uma breve descrição de cada uma dessas classes.

- ***Pair***, esta classe permite adicionar dois objectos, de qualquer tipo, numa só classe. Este facto é muito útil para os casos em que é necessário ter informação emparelhada, como por exemplo o nome de um sensor e o objecto responsável pela manipulação do mesmo.
- ***VideoPlayer***, esta classe manteve-se da Versão 1.0 e permite executar ficheiros vídeo para fornecer o devido *feedback* ao utilizador da aplicação.
- ***AudioPlayer***, tal como a anterior, também esta classe foi herdada da Versão 1.0 e permite reproduzir sons, para ser possível *feedback* alternativo ao do vídeo.
- ***DisplayHandler***, esta classe é a responsável por tudo o que se passa na camada *Multimedia Display*. É ela que executa os vídeos, toca os sons ou altera o fundo consoante o que for programado através da *Authoring Tool*.
- ***XMLHandler***, apesar de não estar visível na arquitectura, esta classe permite a leitura e a escrita dos ficheiros XML. Assim esta classe poder-se-ia localizar como um meio intermediário entre as camadas *Context Processor* e *Data*.
- ***NewSensor***, esta classe, apesar de não aparecer referenciada na arquitectura, é uma classe muito importante para todo o sistema. É através dela que é permitido à *Framework* comunicar com os sensores. O facto de esta classe não aparecer na arquitectura é devido à existência de um objecto da classe *NewSensor* para cada sensor a ser utilizado. Assim, para ser possível uma melhor gestão dos sensores e performance do sistema, os objectos da classe *NewSensor* estão inseridos na classe *SensorPool*.
- ***SensorPool***, esta classe gere todos os objectos necessários à manipulação de cada sensor. É através dela que a camada *Context Processor* consegue comunicar com os sensores externos.
- ***ContextProcessor***, por fim esta classe tem o poder de decisão das acções que o sistema toma. Ou seja, é ela que analisa os dados criados pelo utilizador, no momento da elaboração da aplicação e que os executa em tempo real.

De notar que, para o funcionamento de uma aplicação, não é estritamente necessário a adição de um sensor, ficando esta limitada à execução de ficheiros áudio e vídeo. Os sensores permitem contextualizar a resposta da aplicação em relação ao ambiente envolvente e ao comportamento dos utilizadores.

3.3.2 *Authoring Tool*

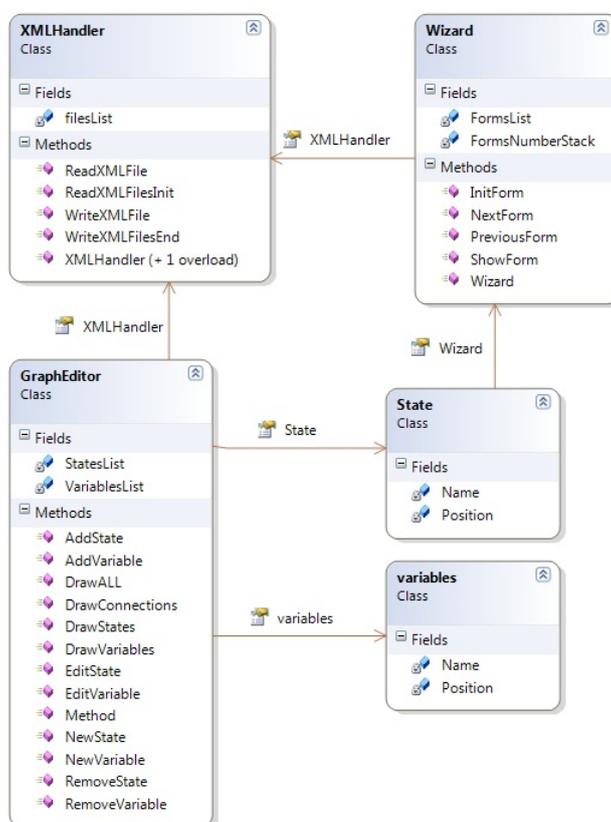


Figura 3.5 Diagrama de classes da *Authoring Tool*

Na figura 3.5, são visíveis cinco classes distintas que se assumem como sendo as mais importantes para o bom funcionamento da *Authoring Tool*. De seguida é apresentada uma breve descrição de cada uma dessas classes.

- **State**, esta classe representa no ambiente gráfico um estado que o utilizador poderá configurar com as acções a determinar. Cada estado possui um *Wizard*, e guarda também a sua posição no grafo, assim como as ligações com outros estados.

- **Variables**, esta classe representa, apenas, a declaração de uma variável, cujo *Context-Processor*, poderá atribuir valores ou não, dependente da programação efectuada pelo utilizador da *Authoring Tool*.
- **XMLHandler**, esta classe permite a leitura e a escrita dos ficheiros XML, que servem de suporte à configuração das aplicações desenvolvidas através da *Authoring Tool*. Assim, esta classe pode ser vista como um meio intermediário entre as camadas *Editor* e *Data*.
- **Wizard**, esta classe responsabiliza-se pela gestão da informação das janelas de configuração que o utilizador preenche de forma a criar uma nova aplicação.
- **GraphEditor**, por fim esta classe é responsável por toda a informação e processamento necessários para que a camada do *Graphical Handler* possa mostrar correctamente todos os aspectos gráficos da aplicação em desenvolvimento. Deste modo é permitido ao utilizador a visualização e a manipulação de uma representação gráfica da aplicação que está a desenvolver.

4. Implementação do sistema

Neste capítulo é descrita a forma de implementação de todo o sistema. Assim proceder-se-á à descrição dos critérios técnicos que foram aplicados e à explicação, de uma forma mais detalhada, de todas as estruturas implícitas no sistema. Deste modo, o capítulo apresenta-se repartido nas seguintes secções:

- **Estruturas de comunicação**, onde serão descritos o sistema de ficheiros que permitem a comunicação entre os Sensores e a *Framework*, assim como os que guardam a informação de modo persistente.
- **Implementação da *Framework***, nesta secção serão apresentados os módulos que a compõem de uma maneira mais pormenorizada.
- **Implementação da *Authoring Tool***, que apresenta os módulos que constituem a *Authoring Tool*.
- **Implementação dos módulos externos**, onde serão descritas, módulo a módulo, as alterações efectuadas e/ou criadas, para que fosse permitida a comunicação com a *Framework*.

4.1 Estruturas de comunicação

Para permitir que qualquer módulo, algoritmo ou mecanismo comunicassem com a *Framework*, de uma forma padronizada e em que o próprio sistema guardasse a informação de um modo persistente, foram desenvolvidos vários ficheiros. Estes utilizam a tecnologia XML para que cada ficheiro seja dotado de uma maneira padrão de explicar a sua arquitectura interna, fazendo-o através de XML SCHEMA. Deste modo, cada tipo de ficheiro, explicado abaixo em maior detalhe, possui um XML SCHEMA que permite aos utilizadores criarem os seus próprios sensores, de forma a estender as capacidades da *Framework*. Assim existem dois ficheiros que são específicos para a comunicação, entre os sensores e a *Framework*, (*Descriptor* e *InOut*) e três para o auxílio da configuração das aplicações desenvolvidas através da *Authoring Tool* (*Sensors*, *Graphs* e *State*).

- **Descriptor**, este ficheiro contém toda a informação necessária para comunicar com um sensor, ou seja é através desta descrição que a *Framework* verifica a forma como comunica com o sensor em questão. De notar que é essencial a existência de um ficheiro *Descriptor*, por cada sensor. Assim, como pode ser verificado através da figura 4.1, o *Descriptor* possui os caminhos dos ficheiros de comunicação, *communicatorPathIN* onde são escritos os *inputs*¹ do sensor e *communicatorPathOUT* onde são escritos os *outputs*² do sensor, nos ficheiros *InOut* descritos no próximo tópico. Existe também um elemento representativo do ficheiro que executa a aplicação, neste caso inicializa o módulo, caso seja necessário. Por fim existe a descrição dos métodos existentes para o sensor. Não havendo limite de métodos é permitido que o sensor disponibilize todas as acções necessárias. Cada método é acompanhado por um ou mais elementos *input* e *output*.

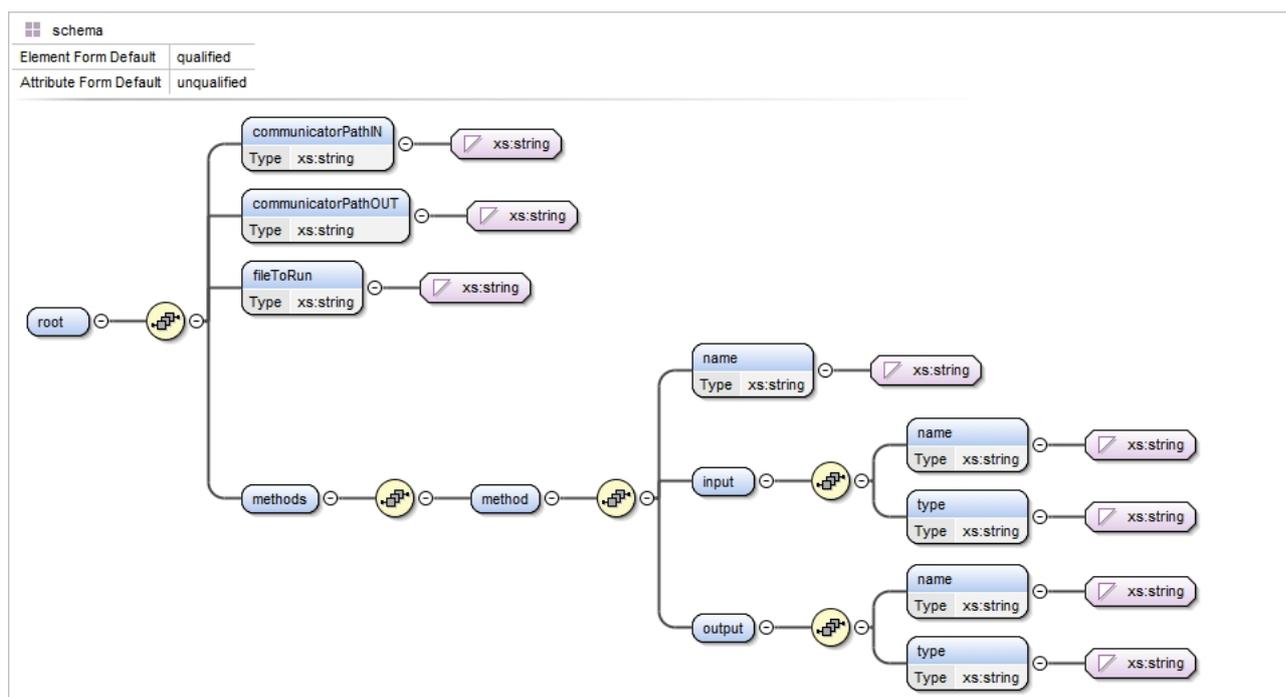


Figura 4.1 Estrutura do XML Schema para os ficheiros que descrevem as propriedades de um sensor

- **InOut**, é um ficheiro de comunicação puro por estar em constante mutação ao longo do tempo de execução de uma aplicação. Como só pretende representar dados de leitura

¹informação enviada para o sensor

²informação escrita pelo sensor

e escrita, este ficheiro é bastante simples, como se pode verificar na figura 4.2. Desta forma, o ficheiro tem um elemento *version* que trata do controlo de versão, permitindo a sincronização das duas extremidades (*Framework* e *Sensor*), o nome do método que está a ser invocado e os respectivos *inputs* ou *outputs*. Tendo em conta que para cada sensor é necessário possuir dois ficheiros *InOut* a figura 4.3, demonstra a relação entre o sensor e a *Framework*, através desses ficheiros.

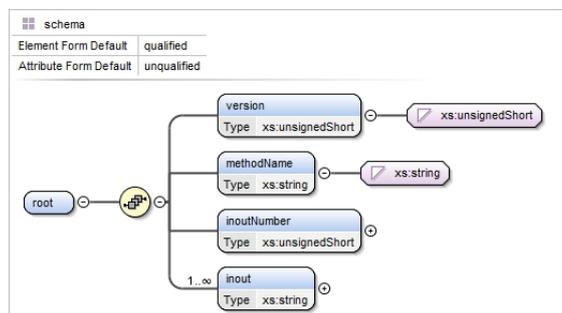


Figura 4.2 Estrutura do XML Schema dos ficheiros de comunicação entre os sensores e a *Framework*

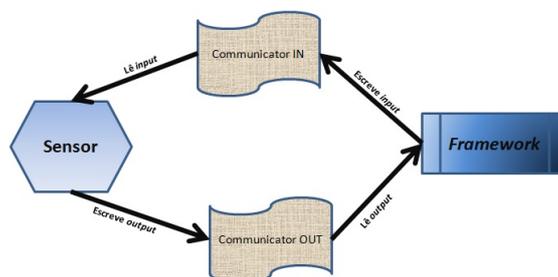


Figura 4.3 Esquema de comunicação através dos ficheiros *InOut*

- *Sensors*, este ficheiro descreve o nome do sensor, atribuído pelo utilizador, e o caminho para o descritor do mesmo (figura 4.4).

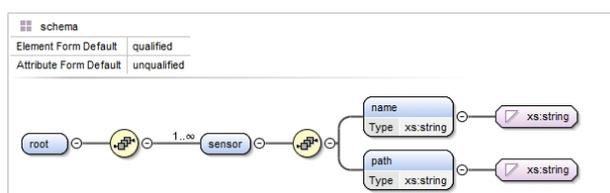


Figura 4.4 Estrutura do XML Schema dos ficheiros que listam os sensores adicionados à *Framework*

- **Graphs**, este ficheiro serve de apoio à *Authoring Tool* no que respeita ao desenho dos estados e das variáveis no seu ambiente gráfico. Possui também a função de informar o sistema sobre os elementos que possam estar presentes em tempo de execução. Desta forma, e através da figura 4.5, é notório que este ficheiro contém a descrição do nome e da posição na aplicação gráfica, tanto para os estados como para as variáveis.

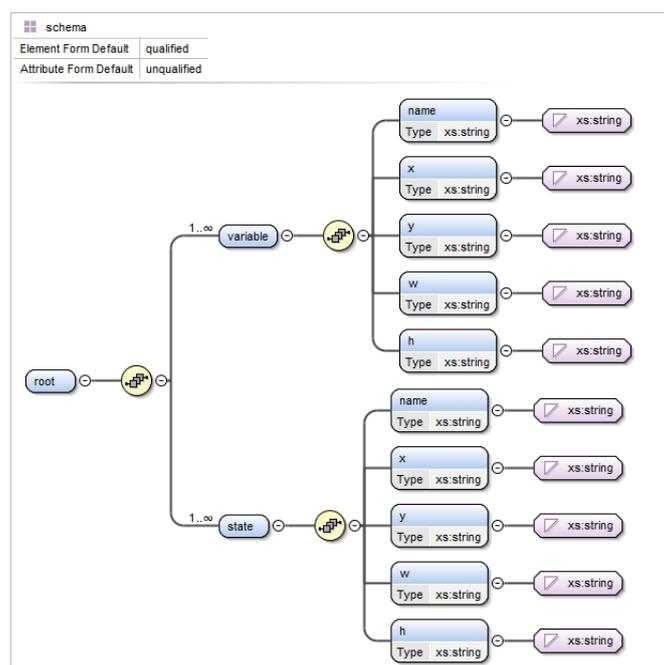


Figura 4.5 Estrutura do XML Schema, dos ficheiros XML que representam os objectos a desenhar na *Authoring Tool*

- **State**, este ficheiro é de longe o mais complexo de todos, pois trata-se do ficheiro que o utilizador configura através da *Authoring Tool*. Assim, através da figura 4.6 é possível ver que este possui sete elementos principais que fazem parte das tarefas mais importantes do *Wizard*, referido na secção 3.2 e descrito em maior detalhe na secção 4.3. Referindo, de cima para baixo, cada elemento tem as seguintes finalidades:
 - *AudioVisual*, primeira acção do estado, poderá tocar um som, reproduzir um vídeo, alterar o fundo ou não executar nenhuma acção;
 - *ChooseAction*, este elemento serve somente para verificar qual a próxima acção, que será uma das quatro seguintes;

- *SendInput*, permite enviar *inputs* para os sensores e no final passar para outro estado;
- *GetOutput*, onde se poderá escolher entre afectar um valor de um *output* a uma variável ou comparar directamente e passar para um outro estado; *Variables*, comparar os valores das variáveis e passar para outro estado;
- *Nothing*, apenas passa para outro estado.
- *Sensor*, que identifica qual o sensor e o respectivo método que está a ser considerado para as acções de *SendInput* e *GetOutput*.

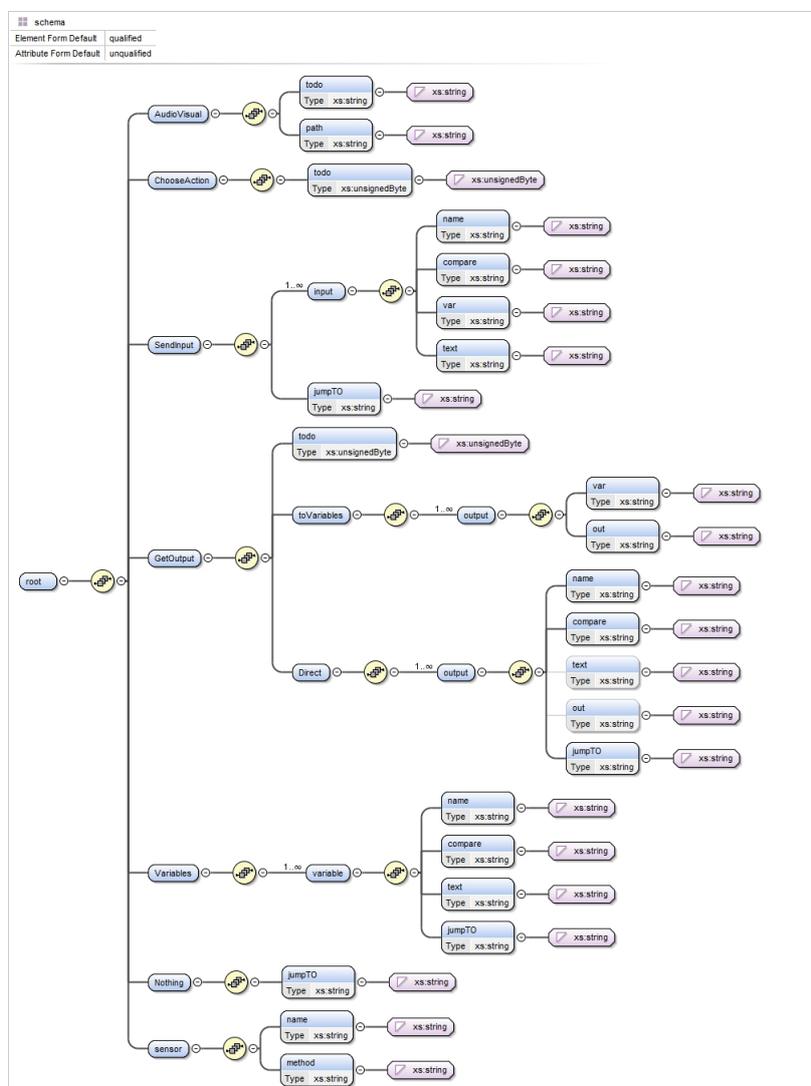


Figura 4.6 Estrutura do XML Schema para os ficheiros que configuram as acções de cada estado

4.2 Implementação da *Framework*

Parte deste trabalho passa pela generalização da *Framework* ao nível dos sensores. Como já foi referido anteriormente, na versão 1.0 existiam no código propriedades específicas para tratar cada sensor, estas restringiam a *Framework* a um número restrito de operações diferentes. Para que todos os tipos de objectos pudessem fazer parte do reportório de sensores a serem levados em consideração, nesta *Framework*, foi desenvolvida uma classe que assumisse a responsabilidade de permitir tal tipo de adição. Desta forma, foi criada a classe *NewSensor*, explicada com maior pormenor adiante, que permite a integração e se encarrega da comunicação entre os sensores e *Framework*. Como também se pretendia criar uma *Framework* mais acessível e versátil, a classe *Context Processor* da versão 1.0 foi alterada como referido na secção 3.1, recorrendo ao potencial que a linguagem C# tem para oferecer.

De seguida são apresentadas as alterações pormenorizadas de cada uma das classes.

4.2.1 *NewSensor*

Esta classe, desenvolvida especificamente para esta nova *Framework*, utiliza as estruturas de comunicação, descritas na secção 4.1, para comunicar com um sensor. Assim, ao ser inicializada, esta necessita de um caminho para o descritor do sensor. Tal facto permite-lhe saber, detalhadamente, onde se encontram os ficheiros de comunicação, a aplicação para ser executada, assim como os nomes dos métodos e dos atributos que cada método possui. Após ter inicializado as suas estruturas internas, visíveis no diagrama de classes (figura 3.4), esta classe disponibiliza quatro métodos que sustentam a comunicação, através dos ficheiros respectivos, com o sensor. Estes métodos são *getInput*, *getOutput*, *sendInput* e *sendOutput*. Desta forma, sempre que estes métodos forem invocados, os ficheiros irão ser modificados de maneira a que ambas as extremidades (sensores e *Framework*) possam capturar os dados. De notar que os métodos de leitura do ficheiro são bloqueantes, ou seja, só se passa à próxima acção depois de, efectivamente, se ter recebido os dados. Este facto é preciso ser levado em conta na construção, através do *Wizard*, das aplicações. Existe ainda um método muito importante que tem o nome de *initSensorAPP*. Este método tem a função de iniciar uma aplicação externa ao sistema, facto que pode ser relevante para aplicações locais. Outro factor a ter em atenção é que existe um objecto desta classe para cada sensor, e esta só consegue tratar de um método de cada vez, podendo ser alterado através do método *setMethodCalled*.

Esta classe está preparada para fazer chamadas remotas. Para tal é necessário que o descritor tenha os endereços remotos das aplicações e/ou ficheiros de comunicação, e conexões devidas para tal comunicação, que não podem ser asseguradas pela *Framework*.

4.2.2 *Context Processor*

A principal alteração que esta classe sofreu foi, sem dúvida, na forma de interpretação de dados, visto que muitas das propriedades da versão 1.0 foram descontinuadas (ex. *Sensor processor*). Desta forma, esta classe tornou-se um pouco mais simplista, pois trata qualquer sensor da mesma maneira, como uma entrada ou saída de dados. De lembrar que o único meio de tratar o *feedback* era através da camada de *Multimedia Display*, visto na secção 3.1, e agora qualquer meio é permitido desde que existam sensores destinados para tratar essas acções. Por exemplo, uma lâmpada que altera a cor consoante a proximidade do utilizador. De notar que esta classe verifica todos os ficheiros referenciados na secção 4.1, através dos quais inicializa as suas estruturas internas.

4.3 Implementação da *Authoring Tool*

Com já foi referido, a *Authoring Tool* serve, exclusivamente, para a criação de aplicações permitindo a utilizadores, que não possuam o conhecimento de programação, desenvolver sistemas persuasivos e tangíveis. Na versão 1.0, o utilizador teria que preencher diversos formulários para configurar as acções da aplicação, podendo somente visualizar um grafo que representava as interacções do sistema. Nesta versão foi introduzida a possibilidade do utilizador manipular o grafo e de configurar cada estado de uma forma normalizada. Esta versão oferece, também, a possibilidade de adicionar um número infinito de estados, facto que não acontecia na versão 1.0, pois esta estava limitada a quinze *Frames*³. Outra opção adicionada foi a possibilidade de se declarar variáveis. Estas são sempre vistas como globais e poderão guardar informação do principio ao fim da aplicação. Este facto é muito útil, por exemplo, se houver a necessidade de guardar o nome do utilizador sem recorrer novamente ao sensor de detecção facial.

É através da *Authoring Tool* que o utilizador consegue configurar os ficheiros descritos na secção 4.1. Para tal, são utilizadas as seguintes classes:

³Intitulados agora por estados

4.3.1 Wizard

Esta classe é a responsável pela definição das acções de cada estado. Ela apresenta várias janelas que permitem a configuração, por parte do utilizador, das várias componentes existentes na *Authoring Tool*. Na imagem 4.7 são visíveis algumas das janelas de configuração utilizadas por esta classe. Segue-se uma pequena explicação, para um melhor entendimento desta classe, sendo que a descrição completa encontra-se no manual de utilizador (apêndice A)



Figura 4.7 Exemplo de configuração através da classe *Wizard*

Na descrição, imediatamente a seguir, cada janela é referida como elemento, visto fazerem parte dos elementos dos ficheiros XML apresentados na secção 4.1, pois cada janela permite editar um ficheiro de configuração.

1. Neste elemento é permitido seleccionar uma das quatro acções básicas que poderão ocorrer em cada estado: mudar o fundo, reproduzir um filme, reproduzir um som e não executar. Nos três primeiros casos é necessário seleccionar o ficheiro correspondente para a acção seleccionada, por exemplo, para alterar um fundo seleccionar uma imagem.
2. Como é possível verificar pela figura, neste elemento só é permitido seleccionar uma acção extra, seja ela enviar ou receber dados, comparar os valores das variáveis ou não executar nenhuma acção e apenas mudar de estado.

3. Este elemento só se torna visível caso, no elemento anterior, tenha sido escolhida uma das opções *Send Input* ou *Get output*. Assim o utilizador pode seleccionar o sensor e o respectivo método a utilizar nas acções seguintes.
4. Quando se selecciona a opção *Get output* no elemento 2, é conveniente que este elemento seja configurado, para que o utilizador possa seleccionar onde colocar os *outputs* provenientes do sensor. Desta forma, poderá seleccionar entre alocá-los em variáveis ou fazer uma comparação directa, como é explicado a seguir.
5. É este elemento que permite ao utilizador, configurar as comparações que são efectuadas aos *outputs* dos sensores. Desta forma pode-se observar na figura que caso o **OUTPUT1** seja igual a **STOP** então irá para o estado **END**⁴, mas se o valor for igual a **A**, então passará para o estado *movie A*.

4.3.2 Elementos gráficos

Os elementos gráficos da *Authoring Tool*, dividem-se em dois, o editor gráfico e os objectos que nele são inseridos. Desta forma temos:

- **GraphEditor**, esta classe é responsável pela representação gráfica da aplicação em desenvolvimento na *Authoring Tool*. É ela a responsável por todo o aspecto gráfico e pela manipulação dos objectos no editor. Para auxiliar os seus cálculos esta classe utiliza os ficheiros de configuração, para verificar as posições iniciais dos estados e das variáveis. Apesar de não ter uma manipulação directa sobre a classe *Wizard*, ela recorre a propriedades dessa classe, unicamente para pedir a cada estado as ligações que estes possuem com outros estados.
- **Variables** e **State**, estas duas classes fazem parte da estrutura de desenho do grafo. Embora a classe *state* guarde o seu *Wizard*, esta serve somente de auxiliar à classe *GraphEditor*, para o apoio do tratamento das imagens dos estados e das suas ligações a outros estados, facto que pode ser observado através da figura 4.8, onde é perceptível a estrutura da aplicação. Assim, os quadrados representam os estados e as linhas a vermelho, as ligações possíveis entre eles. Por fim os rectângulos representam as variáveis declaradas na aplicação a desenvolver.

⁴Estado predefinido como o fim de toda a aplicação

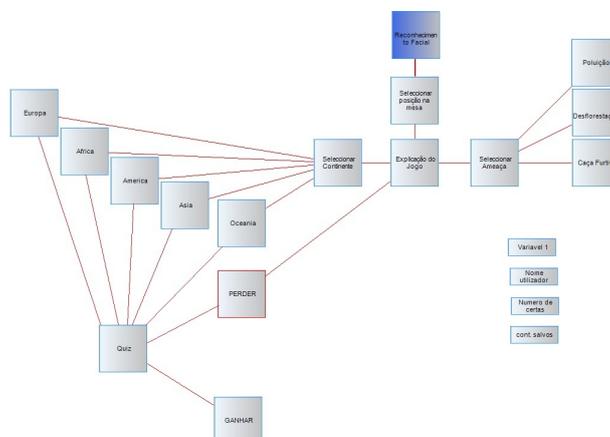


Figura 4.8 Modelo de aplicação desenvolvido na *Authoring Tool*

4.4 Implementação de módulos auxiliares

Ao longo das explicações anteriores, foram referidas várias vezes a existência de sensores, algoritmos ou mecanismo. Consideraremos qualquer um dos três, um módulo independente de toda a *Framework* e respectiva *Authoring Tool* e chamamos-lhe genericamente sensor. Para complementar o que foi descrito anteriormente e demonstrar a generalidade da *Framework*, foram implementadas ou alteradas algumas aplicações que possam ser vistas como sensores. Estas foram implementadas recorrendo também à classe *NewSensor*, pois esta permite fazer a comunicação entre duas extremidades (*Context Processor* e *Sensor*). De seguida segue-se uma breve explicação para cada uma dessas aplicações.

4.4.1 Simulador

Para esta tese foi adaptada uma aplicação⁵, que simula um modelo predador-presa, como descrito na secção 2.6. Esta aplicação encontra-se numa folha de cálculo, visível na figura 2.19 pelo que foi necessário desenvolver um módulo que conseguisse obter e alterar os dados desse mesmo ficheiro. Após o requisito anterior estar completo, apenas foi necessário implementar as comunicações e descrever o módulo através de um ficheiro descritor. Este simulador é bilateral, pois possui métodos de *input*, para receber dados da *Framework* e métodos de *output*, para poder escrever as respostas aos *inputs* recebidos.

⁵<http://home.comcast.net/~sharov/PopEcol/lec10/lotka.html>

4.4.2 Reconhecimento Facial

Para que os utilizadores, na aplicação do caso de estudo, não fossem reconhecidos através de códigos *Data Matrix*, que teriam que carregar num cartão, foi implementado uma solução de reconhecimento facial. Esta solução é uma adaptação de um exemplo, da aplicação *VeriLook 4.0* fornecido pela NeuroTechnology ⁶. Desta forma, a esta aplicação, implementada em C#, foi adicionado a possibilidade de comunicação com a *Framework*, visto que esta ,por si só, já reconhecia a face do utilizador tendo como output o nome do mesmo. De notar que previamente os utilizadores teriam de fazer uma subscrição na base de dados, onde era obrigatório fornecer os dados do utilizador (nome e fotografia). Este módulo possui, apenas, métodos de *output*, onde é escrito o nome do utilizador reconhecido pelo sistema implementado.

4.4.3 Leitura de *Data Matrix*

Este módulo foi inteiramente aproveitado da versão 1.0, lembrando que este divulgava o código que era capturado por uma câmara. Assim, sempre que um código fosse capturado era enviado para o ficheiro de comunicação (*inOut*), até que a *Framework* necessitasse de o ler. Este módulo possui apenas métodos de *output*, onde são escritos os códigos *Data Matrix* reconhecidos pela aplicação.

4.4.4 Girador e Sensores

Apesar de serem módulos distintos, estes foram criados para trabalhar em conjunto. O Girador trata-se de uma aplicação que roda o ecrã, mediante os graus, múltiplos de noventa, que lhe forem enviados por *input*. Por sua vez, o módulo Sensores é composto por quatro sensores de infravermelho, que consegue detectar a presença a cerca de sessenta centímetros de um utilizador, pelo que poderá verificar em que parte da mesa (do caso de estudo) este se encontra, escrevendo a posição no ficheiro de comunicação (*InOut*), através do método de *output* que possui. Desta forma, através da *Framework* os módulos podem complementar-se, uma vez que um verifica a posição do utilizador e o outro roda o ecrã para essa mesma posição.

⁶http://www.neurotechnology.com/v1_sdk.html

4.4.5 *TextToSpeech*

Este módulo tinha a capacidade de interpretar os *inputs*, compostos exclusivamente por texto, e traduzir para um discurso áudio, através da tecnologia *text-to-speech*⁷. Este módulo serviria, para analisar o nome do utilizador, capturado pelo módulo de reconhecimento facial, de forma a poder personalizar a interacção. Mas como a reprodução áudio deste módulo não se apresentava de uma forma clara, não sendo perceptível o nome de alguns utilizadores, este não foi adicionado no caso de estudo.

⁷<http://msdn.microsoft.com/en-us/library/cc998523.aspx>

5. Caso de Estudo

Com o fim de testar a *Framework* descrita nos capítulos anteriores, foi elaborado através da utilização da *Authoring Tool*, um sistema que tem como base de interacção as interfaces tangíveis. Esta aplicação intitula-se *Ecosystem Room*. Esta tem como principal objectivo consciencializar as crianças para o equilíbrio do ecossistema e para as ameaças que neste existem, tal como a poluição, a caça furtiva e a desflorestação. Através da consciencialização para os problemas ambientais, pretende-se inculcar nos utilizadores novos comportamentos que contribuam para melhorar as condições do meio ambiente. O estudo subjacente à construção deste sistema é comparar a potencialidade das interfaces tangíveis em relação às interfaces gráficas convencionais, na problemática da aprendizagem e persuasão, através de meios computadorizados. Desta forma, foram desenvolvidas duas versões do *Ecosystem Room* que diferem na forma de interface com o utilizador: Uma possui uma interface tangível e a outra uma interface gráfica convencional em que a interacção é efectuada através de periféricos, como por exemplo o rato. Assim, para cada objecto físico presente no sistema tangível, existe um objecto na interface gráfica, e em vez de este ser colocado em cima da mesa, como acontece no sistema tangível, é arrastado a representação do objecto para o centro da área de jogo, utilizando o rato. O reconhecimento facial, para identificação do utilizador, é efectuada de forma idêntica para ambos os casos.

As secções seguintes contêm a descrição do caso de estudo de uma forma mais detalhada, apresentando-se pela seguinte ordem.

- **Desenvolvimento do caso de estudo**, nesta secção é descrita a arquitectura do sistema e apresentado cada elemento que a compõe.
- **Avaliação da *Framework* e respectiva *Authoring Tool***, nesta secção descrevem-se os testes e demonstram-se os resultados da avaliação efectuada à *Framework* e à *Authoring Tool*.
- **Avaliação do *Ecosystem Room***, demonstração dos resultados segundo os testes, efectuados com crianças, à aplicação desenvolvida, tanto a tangível como a gráfica convencional.

5.1 Desenvolvimento do caso de estudo

Nesta secção encontra-se a descrição do *Ecosystem Room* quer ao nível das componentes físicas de *hardware* quer ao nível de *software*, incluindo as regras do jogo e os objectivos que os utilizadores devem atingir. Numa primeira abordagem, deve ser referenciado que antes da construção deste sistema foi estudado um outro protótipo. Este era baseado num jogo de perguntas e respostas cujo utilizador (uma criança) poderia usar como auxiliar aos estudos. Todavia, este protótipo mostrou-se demasiado simplista e pouco adequado ao estudo que se pretende efectuar e, desta forma, partiu-se para uma abordagem em que a persuasão do sistema e a aprendizagem do utilizador fossem mais valorizados.

O *Ecosystem Room* é um jogo, que tem como objectivo ensinar o que é um ecossistema e consciencializar, as crianças, para os problemas que o ameaçam. Este jogo é vocacionado para crianças do primeiro ciclo do ensino básico, visto estas ainda não terem noções concretas do que é um ecossistema e quais as suas ameaças. Durante o jogo o utilizador é incitado a salvar vários ecossistemas, sendo que cada um possui uma ameaça diferente. Desta forma, pretende-se que o utilizador fique a saber o que é um ecossistema, quais as suas ameaças e quais os métodos de as combater. Os pormenores do jogo, encontram-se descritos na secção 5.1.2.

5.1.1 Arquitectura do *Ecosystem Room*

O *Ecosystem Room* trata-se de um sistema, que se encontra isolado numa sala, onde o utilizador é incentivado a entrar para que possa aprender mais sobre o funcionamento dos ecossistemas. Assim, pode-se observar na figura 5.1, todo o ambiente envolvente ao sistema. A zona de entrada do *Ecosystem Room* (*Ecosystem Room Entrance Hall*) encontra-se separada do resto do ambiente. Ao entrar no *Ecosystem Room Entrance Hall*, o utilizador depara-se com um monitor, que se encontra junto da marcação dois (2), que reproduz um filme sobre um ecossistema. Aquando da visualização do filme pelo utilizador, uma câmara de vídeo, que se encontra na parte superior do monitor, captura imagens do utilizador e envia para o sensor de reconhecimento facial, que o identifica e comunica à *Framework*. Desta forma, após ser identificado, o utilizador é incentivado a prosseguir para a sala de interacção através de instruções áudio.

Ao entrar na sala de interacção, o utilizador depara-se com uma mesa interactiva central onde decorre toda a acção do jogo. Na figura 5.2 encontra-se um esquema com a descrição pormenorizada da mesa interactiva. A figura 5.3 mostra a mesa que foi criada para este sistema,

neste caso a ser utilizada por uma criança durante os testes de avaliação.

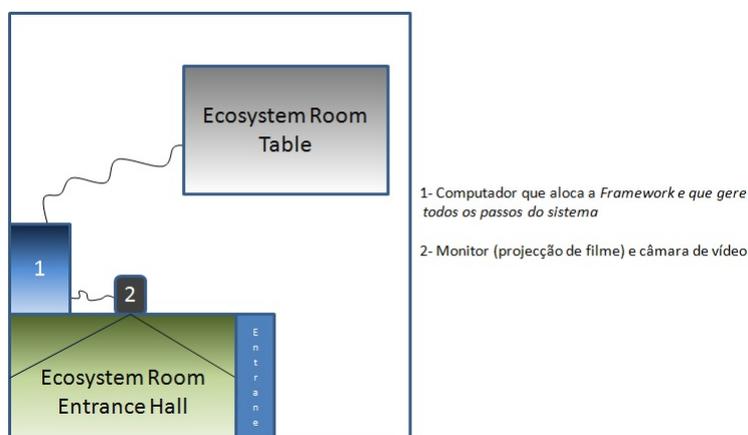


Figura 5.1 Ambiente de *Ecosystem Room*

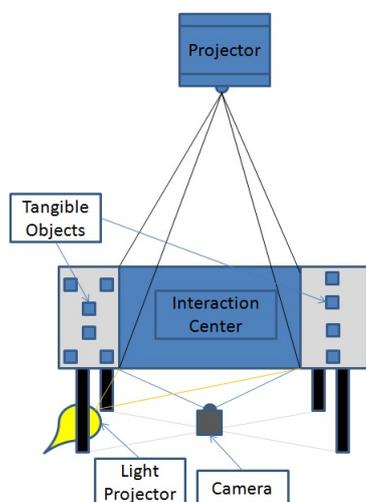


Figura 5.2 Mesa de *Ecosystem Room*



Figura 5.3 Criança a interagir com *Ecosystem Room Table*

Segue-se a descrição detalhada de cada elemento que compõe a mesa interactiva.

- **Projector**, este é o elemento responsável pela projecção dos gráficos na área de interacção.
- **Interaction Center**, este elemento possui uma folha de papel vegetal sobre o tampo da mesa, que é composto por um segmento de acrílico transparente. Desta forma é permitida

a projecção do conteúdo no topo da mesa, sem que a transparência do acrílico faça interferência. Esta transparência é essencial para a detecção dos códigos *Data Matrix* situados na base dos objectos tangíveis.

- ***Tangible Objects***, são os objectos fornecidos ao utilizador para que este possa interagir com o sistema. Cada objecto é composto por uma camada superior (topo), com um desenho alusivo à sua função e uma camada inferior (base) com o código *Data Matrix* a ser reconhecido pelo sistema. Sendo a mesa transparente, e possuindo uma folha de papel vegetal sobre esta, quando um objecto é colocado sobre ela este torna-se visível para a câmara que se encontra por baixo, permitindo a detecção e reconhecimento do código *Data Matrix* que se encontra na base do objecto (figura 5.4).
- ***Camera***, este elemento representa um sensor de leitura de códigos *Data Matrix*.
- ***Light Projector***, este elemento foi utilizado para aumentar a luminosidade na parte inferior da mesa, permitindo melhorar a detecção do código por parte da câmara.



Figura 5.4 Visão inferior da *Ecosystem Table*

Para permitir comparar os dois tipos de interfaces, foi desenvolvida também uma aplicação, com uma interface gráfica convencional, que simulasse a acção que se passava no topo da mesa,

tal como é visível na figura 5.5. Deste modo, para cada acção na interface tangível existe uma acção correspondente na interface gráfica. Para a utilização desta interface é substituída a mesa, por um computador com o dispositivo adequado à manipulação desta interface (rato).

Os filmes e imagens utilizados no *Ecosystem Room* foram criados em *Flash* e todas as explicações foram gravadas com a voz de uma criança de dez anos. Por sua vez as imagens e os personagens foram adaptados de outros filmes conhecidos. Desta forma, o filme torna-se mais alusivo às idades do público-alvo, entre os oito e os dez anos de idade.



Figura 5.5 Visão da interface gráfica convencional

5.1.2 Regras e procedimentos

Como referenciado anteriormente, o *Ecosystem Room* é um sistema que tenta consciencializar as crianças para os problemas relacionados com os ecossistemas. Para ser possível abordar a criança de uma maneira não intrusiva e apelativa, este sistema assume um conceito de jogo, para que o utilizador possa aprender num contexto de brincadeira. Este jogo obedece a um determinado número de regras e a alguns procedimentos bem definidos, que são apresentados ao utilizador no início do jogo.

Ao iniciar o jogo é apresentado ao jogador, através de um vídeo alusivo, uma explicação sobre um ecossistema, neste é referida a definição de ecossistema e aclarada cada uma das

suas componentes. Após isso, são apresentados, também por vídeo, algumas das ameaças que podem interferir com um ecossistema (ex: Caça furtiva, Desflorestação, Poluição, Urbanismo, Espécies em vias de extinção). Após estas duas explicações, o utilizador é incitado a utilizar o sistema. Assim, este poderá seleccionar, utilizando os objectos tangíveis, um de cinco filmes, cada um com a explicação de uma ameaça, ou então iniciar a aventura para salvar o ecossistema. Caso este seleccione um filme, então irá assistir à reprodução deste, voltando no final, para o menu anterior. Caso seleccione iniciar a aventura, então este será incentivado a salvar vários ecossistemas existentes no mundo. Como é mostrado na tabela 5.1, para cada continente existe uma ameaça diferente, sendo que o utilizador deve tentar salvar todos os continentes. Para tal, este selecciona um continente para salvar, colocando o objecto correspondente ao continente no centro da mesa, entrando num nível do jogo em que terá de responder acertadamente a pelo menos duas de três perguntas, que têm como tema a ameaça correspondente ao continente que o utilizador seleccionou. Caso o utilizador consiga responder acertadamente a pelo menos duas perguntas, então o continente ficará a salvo, caso contrário este é incentivado a tentar novamente. Quando o utilizador salvar todos os continentes, então o mundo ficará a salvo e o utilizador ganha o jogo.

Continente	Ameaça
África	Caça Furtiva
América	Desflorestação
Ásia	Poluição
Europa	Urbanismo
Oceânia	Espécies em vias de extinção

Tabela 5.1 Mapeamento das ameaças com os continentes

Neste jogo existem duas particularidades a tomar em consideração:

- **Nunca se perde**, este jogo foi construído para que um utilizador nunca perca, assim este poderá não acertar nas perguntas e não salvar o continente de imediato, mas é incentivado a tentar novamente, podendo desta forma aprender mais sobre a ameaça de cada continente. De notar que é um jogo vocacionado para a aprendizagem de uma matéria escolar e que visa contribuir para melhorar o comportamento ambiental das crianças desde muito novas.
- **Utilização do simulador**, o módulo de simulação apresentado no capítulo 4, é utilizado

nas perguntas relacionadas com as espécies em vias de extinção. É simulado um modelo predador-presa, onde as presas são espécies em vias de extinção, demonstrando ao utilizador a importância da conservação das espécies. As crianças utilizando objectos tangíveis podem manipular os animais dos diferentes níveis da cadeia alimentar, visualizando as relações entre as diferentes espécies, tomando consciência para a importância de cada uma delas.

5.2 Avaliação da *Framework* e respectiva *Authoring Tool*

Para avaliar as capacidades da *Framework* e a usabilidade da *Authoring Tool* na construção de sistemas persuasivos sensíveis ao contexto, foram realizados alguns testes envolvendo professores do primeiro, segundo e terceiro ciclos de escolaridade. Estes foram incitados a desenvolver pequenas aplicações, para poderem dar o seu parecer sobre a aplicação. De seguida são apresentados estes testes e os respectivos resultados.

5.2.1 Participantes

Os participantes destes testes de avaliação foram exclusivamente professores de todos os ciclos do ensino básico, uma vez que são parte do público-alvo desta *Authoring Tool*. Assim, participaram ao todo dez (10) professores, cinco (5) do primeiro ciclo, três (3) do segundo e dois (2) do terceiro.

5.2.2 Metodologia

Inicialmente a aplicação foi apresentada aos participantes tendo sido explicado o objectivo e o funcionamento da mesma. De seguida, cada participante teve a oportunidade de utilizar a aplicação para construir duas simples aplicações. Assim, a cada participante foram apresentados dois cenários, que serviram de base para a construção das aplicações, que são descritos a seguir:

1. Inicializar o sistema com um vídeo, à escolha entre três pré-definidos, e finalizar num outro estado com o som de "PARABÉNS".
2. Inicializar o sistema com um fundo com três filmes, sendo que o fundo e os filmes são pré-definidos. A aplicação deve permitir ao utilizador seleccionar o filme que quer visualizar

ou terminar a aplicação. Isto através da colocação do respectivo objecto, que contém o código *Data Matrix*, sobre a mesa.

Durante as diversas fases do decorrer dos testes os participantes foram questionados de forma a expressar a sua opinião acerca da aplicação em estudo:

- Antes da interacção com a *Authoring Tool* os utilizadores foram questionados, sobre a possibilidade de eles próprios criarem um sistema computadorizado, para leccionarem os conteúdos nas aulas.
- Ao longo da interacção os utilizadores foram incentivados a revelar quais as suas dificuldades, e a apontar potenciais alterações na ferramenta.
- Por fim, após a conclusão do teste, os utilizadores foram novamente questionados sobre a possibilidade de eles utilizarem a ferramenta para os conteúdos das aulas e sobre a facilidade de uso da *Authoring Tool*.

Na secção seguinte são apresentados os resultados das entrevistas efectuados a cada utilizador.

5.2.3 Apresentação de resultados

Uma vez que as questões colocadas aos participantes foram questões abertas, os resultados não são apresentados de forma gráfica, mas sim descrevendo o que foi apontado à aplicação. Os resultados são apresentados de seguida em três tópicos, correspondendo cada um a uma fase da execução dos testes.

- **Antes da Interacção**, os utilizadores estavam de certa forma relutantes em aderirem à ideia de eles próprios elaborarem um sistema computacional. Para além de acharem que teriam de possuir conhecimentos de programação, achavam que teriam mais dificuldade em preparar os conteúdos.
- **Ao longo da Interacção**, os utilizadores mostraram ter algumas dificuldades na compreensão de alguns aspectos técnicos, como por exemplo na designação de variáveis os nomes *String* e *Integer*, não faziam sentido para alguns. No entanto, todos os utilizadores terminaram com sucesso a execução dos dois cenários. Apenas dois deles necessitaram de ajuda extra na configuração do ficheiro, através do *Wizard*. De uma forma

geral, mostraram-se satisfeitos tanto com o ambiente gráfico, como com a simplicidade da criação uma aplicação interactiva.

- **Após a Interação**, os utilizadores em geral mostraram-se satisfeitos por terem aprendido a criar um sistema, e mostraram-se entusiasmados com o facto de poderem criar, eles próprios, um sistema interactivo para educação. Um dos professores do primeiro ciclo referiu que poderá ser bastante interessante utilizar nas aulas na aprendizagem do abecedário, pois a criança ao manipular a própria letra teria mais facilidade de assimilar a sua forma e relacioná-la ao seu som.

Tendo em conta o interesse que os utilizadores demonstraram, em relação ao sistema, este foi requisitado pelo Colégio Atlântico para ser disponibilizado aos professores e educadores para que estes possam criar aplicações lúdicas para os seus alunos. Desta forma, na semana cultural (de 22 Março de 2010 a 26 Março de 2010), existirá um dia dedicado às T.I.C.¹, onde o sistema será demonstrado aos professores, educadores, alunos e pais.

5.3 Avaliação do *Ecosystem Room*

Para avaliar o sistema *Ecosystem Room*, foram realizados alguns testes com crianças. Onde era pretendido testar a usabilidade da aplicação tangível e comparar as duas interfaces em termos de usabilidade, aprendizagem e efeito persuasivo. De seguida, descreve-se pormenorizadamente os testes efectuados.

5.3.1 Participantes

Os participantes destes testes de avaliação foram exclusivamente crianças do primeiro ciclo do ensino básico. Participaram nos testes trinta crianças com idades compreendidas entre os oito e os dez anos (figura 5.6). Vinte e um eram do sexo masculino e nove do sexo feminino (figura 5.7).

¹Tecnologias da Informação e Comunicação

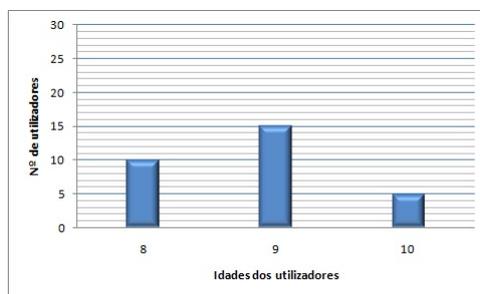


Figura 5.6 Distribuição da idade dos utilizadores; média = 8.83; desvio padrão = 0.69

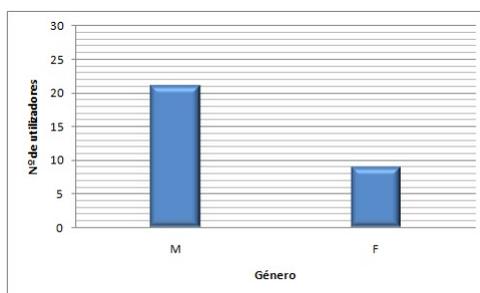


Figura 5.7 Distribuição do sexo dos utilizadores

5.3.2 Metodologia

Os utilizadores foram incitados a entrar para o *Ecosystem Room*, aos pares, sendo que um iria utilizar a aplicação tangível e outro a aplicação gráfica. De notar que cada utilizador entrou sempre duas vezes na *Ecosystem Room*, uma para utilizar a aplicação gráfica, outra para utilizar a aplicação tangível, sendo questionados uma vez no final de cada jogo. Metade dos participantes utilizou primeiro a aplicação tangível e a outra metade utilizou primeiro a interface gráfica convencional.

Durante a realização dos testes procedeu-se à observação dos utilizadores enquanto estes interagiam com o sistema e eram incentivados a expressar o seu raciocínio. Por sua vez, o observador ia anotando as observações num formulário criado para o efeito. Os utilizadores foram questionados em três fases, uma antes de qualquer interacção, uma após a interacção com a primeira interface e por fim após a interacção com o segundo tipo de interface. O questionário foi formulado de maneira a facilitar o seu preenchimento pela criança. Foram utilizados elementos descritos na secção 2.5, tal como o *smileyometer*, para verificar o grau de satisfação do utilizador, ou o *Fun Sorter*, para verificar qual das aplicações os utilizadores preferiram.

Os participantes foram também questionados sobre a temática em questão, antes e depois dos testes para que se pudesse tirar elações sobre a aprendizagem de conteúdos desconhecidos. De seguida, serão apresentados os resultados obtidos, após análise cuidada dos questionários.

5.3.3 Apresentação de resultados

De seguida são apresentados os resultados obtidos através da análise cuidada de cada questionário. No final serão feitas alusões a alguns aspectos de observação da usabilidade da aplicação.

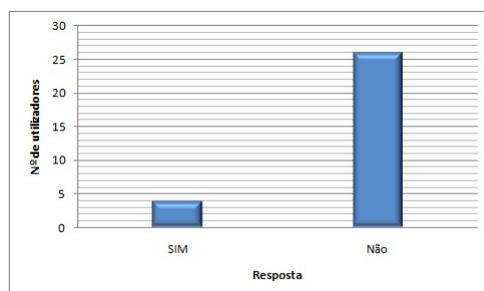


Figura 5.8 Número de utilizadores que sabem ou não o que é um ecossistema

Antes de qualquer interacção foi perguntado a cada utilizador se sabia o que era um ecossistema, pelo que a grande maioria respondeu não, tal como podemos verificar na figura 5.8. A situação é normal, visto a maioria dos utilizadores ainda não ter sido abordado com esta temática na escola. Foi também perguntado aos utilizadores se sabiam o que eram espécies em vias de extinção, um problema grave que pode comprometer o equilíbrio dos ecossistemas. Vários utilizadores já tinham a percepção do que são espécies em vias de extinção, tal como é demonstrado na figura 5.9. Na sequência da questão anterior, os utilizadores foram interrogados sobre que espécies conheciam que estivessem efectivamente em vias de extinção. Como se verifica na figura 5.11, estes conhecem vários animais que efectivamente estão em perigo de extinção. Na sequência da pergunta anterior, os utilizadores foram questionados sobre as causas que podem provocar a extinção das espécies, e sobre o que poderiam fazer para combater o problema. Assim, pode ser observado na figura 5.10, que estes conheciam algumas das causas. No entanto não manifestaram conhecimento acerca de possíveis soluções. Seguidamente, ao serem questionados sobre se sabiam que os animais em vias de extinção constituem uma ameaça ao ecossistema, nenhum dos utilizadores respondeu que sim. Por fim antes do início de qualquer interacção, foi pedido aos utilizadores para indicarem o grau de motivação para participar no

jogo. Os resultados podem ser vistos na figura 5.12. Para responder a esta pergunta, as crianças utilizavam um *smileyometer* cuja escala variava entre 1(um, extremamente desmotivado) e 5(cinco, extremamente motivado).

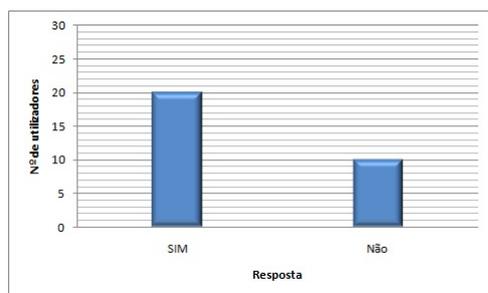


Figura 5.9 Número de utilizadores que sabem ou não o que é uma espécie em vias de extinção

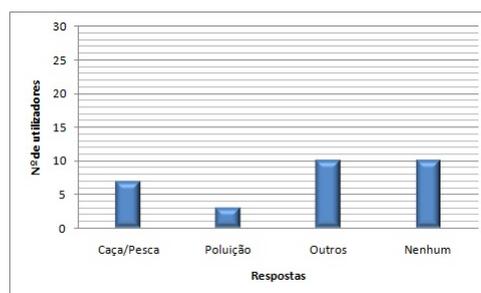


Figura 5.10 Causas que podem provocar a extinção de animais

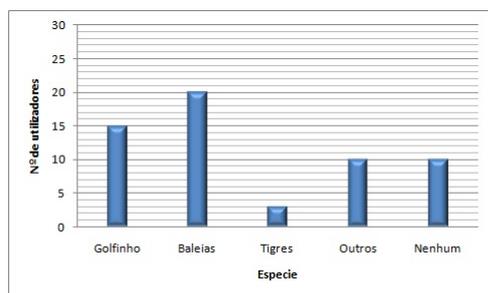


Figura 5.11 Número de animais que os utilizadores conhecem e que estão em vias de extinção

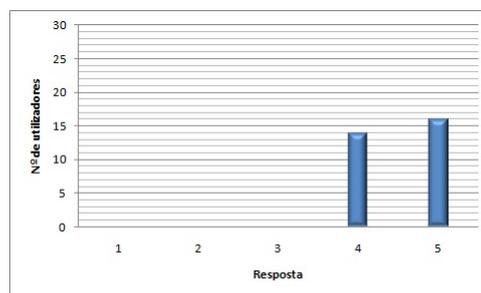


Figura 5.12 Motivação dos utilizadores antes das interacções; média=4.53; desvio padrão = 0.5

No final das interacções os utilizadores mostraram-se bastante satisfeitos por terem jogado (figura 5.14), e indicaram ter gostado bastante do jogo (figura 5.13). Estes mostraram ainda ter assimilado alguns conhecimentos sobre os ecossistemas, embora não fossem cientificamente correctos nas respostas.

Em termos de comparação das duas interfaces, foi notória a preferência pela interface tangível, figura 5.15. Os participantes também indicaram preferir a interface tangível quando questionados sobre qual a que gostariam de utilizar para aplicações de estudo, ou seja para efeitos de aprendizagem, figura 5.16.

Foram também anotadas algumas observações efectuadas durante as diversas fases de interacção dos participantes com a aplicação desenvolvida

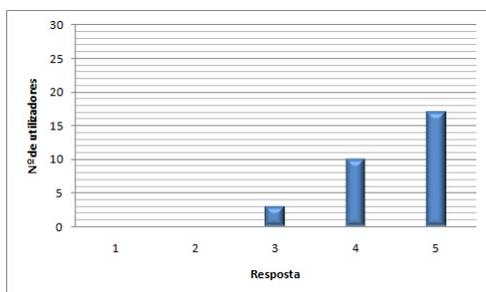


Figura 5.13 Satisfação do utilizador em relação ao jogo; média = 4.47; desvio padrão = 0.68

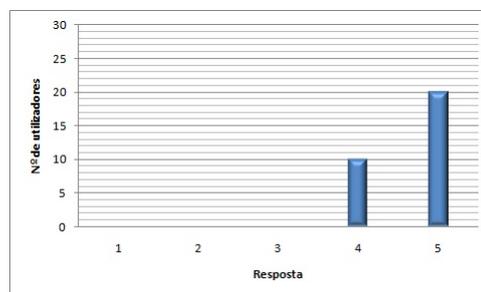


Figura 5.14 Satisfação do utilizador após ter jogado; média = 4.67; desvio padrão = 0.48

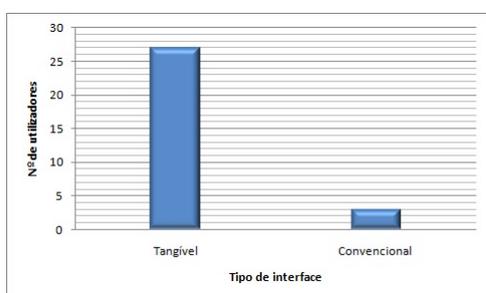


Figura 5.15 Preferência da interface, em relação ao jogo

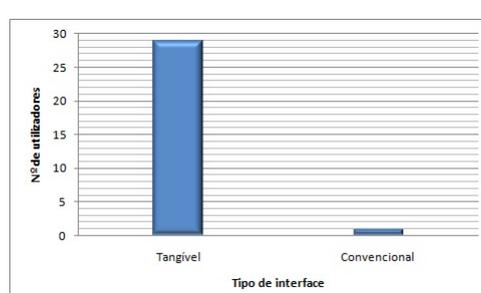


Figura 5.16 Interface preferida para outros contextos educativos

- **Identificação do utilizador**, como se pode verificar na figura 5.17 todos os utilizadores conseguiram completar sem dificuldades este passo.
- **Compreensão do jogo/interacção**, nesta fase é pretendido avaliar a facilidade com que os participantes compreendem as regras do jogo e a forma de interacção. Deste modo obtiveram-se os resultados apresentados na figura 5.18, onde se verifica que nem todos os utilizadores conseguiram compreender o que era requerido. Este facto aconteceu, por dois motivos, assinalados nos campo de observação de cada questionário, que são: a baixa qualidade do som, tanto do altifalante como da gravação e a dicção do locutor que por vezes não era a mais correcta.
- **Interacção com a mesa durante o jogo**, nesta fase pretendeu avaliar-se a facilidade de interacção com a mesa (e os objectos tangíveis) durante o jogo. Deste modo, foram obtidos os resultados apresentados na figura 5.19, onde se verifica que a interacção foi efectuada sem dificuldades, para a maioria dos utilizadores. Algumas dificuldades encontradas pelos utilizadores ficaram a dever-se a problemas na captura dos códigos *Data*

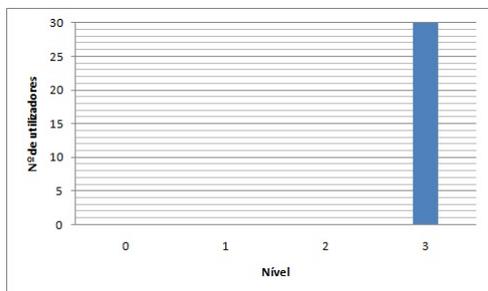


Figura 5.17 Resultados da identificação do utilizador; média = 3; desvio padrão = 0

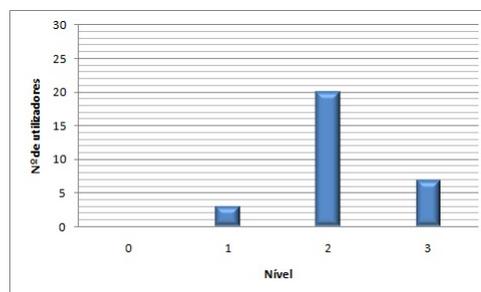


Figura 5.18 Resultados da compreensão do jogo/interacção; média = 2.13; desvio padrão = 0.57

Matrix pela câmara por causa da falta de luminosidade. Os utilizadores manifestaram bastante entusiasmo e interesse em interagir com a mesa e o *Ecosystem Room*.

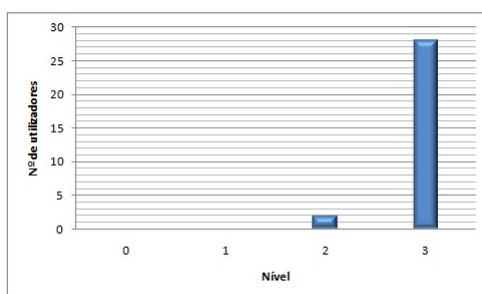


Figura 5.19 Resultados da interacção com a mesa durante o jogo; média = 2.93; desvio padrão = 0.25

5.3.4 Considerações globais

Após as interacções com cada um dos sistemas (interface tangível e interface gráfica convencional), os utilizadores foram questionados sobre os conteúdos que lhes haviam sido apresentados, e de uma forma geral todos conseguiram responder acertadamente às perguntas, não sendo por isso possível verificar se para efeitos de aprendizagem, num curto espaço de tempo, qual a interface mais apropriada. Mas verifica-se que os utilizadores preferem utilizar o sistema tangível, visto poderem utilizar a própria mesa para estudarem e serem ajudados pelo sistema em simultâneo. Verificou-se também uma maior sensibilização das crianças em relação aos problemas ambientais, tratados no *Ecosystem Room*, tendo estas ficado alertadas para as causas e soluções possíveis para estes problemas, bem como para a necessidade da sua contribuição.

Durante a realização dos testes, uma turma de vinte e um alunos do nono ano de escolaridade, com idades compreendidas entre os catorze e quinze anos. Foram questionados e testaram também as aplicações. Apesar de não serem contabilizados para os resultados apresentados anteriormente, foi unânime, entre os alunos, que o sistema tangível era de facto mais interactivo, e mais cativante, embora possa tornar-se menos prático, para aplicações mais complexas, por exemplo: edição de imagem, escrita de documentos ou apresentações. Mas que para o estudo e para jogos (pedagógicos ou lúdicos), esta aplicação consegue para além de cativar mais o utilizador, de fazer com que este se concentre mais na acção do jogo, e nas instruções que lhe são dadas, melhorando o seu empenho na execução das tarefas.

6. Conclusão da dissertação e trabalho futuro

Com a evolução da tecnologia, os sistemas que conseguem ser cientes do contexto do meio que os envolve tornam-se cada vez mais úteis para propósitos persuasivos. Estes sistemas permitem, de forma não intrusiva, detectar as condições do ambiente envolvente e as acções de utilizadores de forma a responder de acordo com elas e influenciar o comportamento desses mesmos utilizadores. A aprendizagem e a consciencialização, nomeadamente em relação a problemas ambientais, permite aos utilizadores adoptarem comportamentos mais adequados à resolução (ou não agravamento) desses problemas.

6.1 Conclusão

Este trabalho juntou tecnologia persuasiva com as interfaces tangíveis para criar uma nova visão de interacção com os utilizadores. Para tal, foi desenvolvida uma *Framework* para criar e executar aplicações persuasivas, desenvolvidas por utilizadores sem conhecimentos em programação, permitindo o uso de interfaces tangíveis. Esta *Framework* tem como característica principal o facto de ser genérica, pois permite adicionar qualquer tipo de sensor físico, algoritmo ou mecanismo, sem necessitar de alterar a sua estrutura interna. Esta mostrou-se bastante eficaz nos requisitos impostos, visto permitir executar, sem problemas, as aplicações desenvolvidas através da *Authoring Tool*. Esta última foi criada para que os utilizadores, sem conhecimentos de programação, pudessem desenvolver aplicações persuasivas sensíveis ao contexto e recorrendo ao uso de interfaces tangíveis.

Foram efectuados testes com alguns utilizadores, de forma a verificar a usabilidade da *Authoring Tool*. Nesse estudo, alguns utilizadores apresentaram dificuldades na compreensão de aspectos intrínsecos da ferramenta, que foram corrigidos após a realização dos testes, mas em geral os utilizadores acharam esta ferramenta de fácil uso. Deste modo, alguns utilizadores, visto serem exclusivamente professores, pretendiam utilizar este sistema para fazer a abordagem de conteúdos escolares aos seus alunos. referiram ainda, que poderia trazer vantagens na aprendizagem de conceitos básicos de uma forma mais eficaz, uma vez que captava o interesse dos alunos. Como por exemplo, na aprendizagem do abecedário, pois a criança ao manipular a própria letra teria mais facilidade de assimilar a sua forma e relacioná-la ao seu som.

Tendo em conta o interesse que os utilizadores demonstraram, em relação ao sistema, este

foi requisitado pelo Colégio Atlântico para ser disponibilizado aos professores e educadores para que estes possam criar aplicações lúdicas para os seus alunos. Desta forma, na semana cultural (de 22 Março de 2010 a 26 Março de 2010), existirá um dia dedicado às T.I.C.¹, onde o sistema será demonstrado aos professores, educadores, alunos e pais.

No decorrer deste trabalho, para funcionar como caso de estudo, apresentado no capítulo 5, foi desenvolvido, através da *Authoring Tool*, um sistema focalizado na consciencialização para a problemática dos ecossistemas e respectivas ameaças. Intitulado de *Ecosystem Room*, esta aplicação, vocacionada para crianças do primeiro ciclo do ensino básico, foi desenvolvida para demonstrar às crianças o que são os ecossistemas e quais os perigos que os ameaçam. Esta aplicação apresenta características persuasivas, incitando o jogador a entrar na sala, para jogar, na apresentação dos conteúdos de uma forma apelativa, que faz com que o utilizador apreenda os conteúdos que são divulgados ao longo da interacção, com a intenção de alterar a sua atitude em relação ao meio ambiente.

Esta aplicação foi desenvolvida, também, para estudar os efeitos persuasivos na aprendizagem que as interfaces tangíveis têm face às interfaces gráficas convencionais. Deste modo, o *Ecosystem Room* possui duas versões, cada uma com um meio de interacção distinto, que são:

- Através de interfaces tangíveis, onde é utilizada uma mesa interactiva e alguns objectos tangíveis, recorrendo à tecnologia de reconhecimento de códigos *Data Matrix*;
- Através de interfaces gráficas convencionais, tendo como meio de interacção o rato.

Para avaliar esta aplicação foram efectuados testes com trinta utilizadores. Desta forma foi possível concluir que numa primeira abordagem alguns dos utilizadores, que não conhecem a manipulação de interfaces tangíveis, ficam inicialmente relutantes em relação à sua utilização. Mas ao interagir com uma aplicação tangível, nota-se que a maioria das crianças prefere utilizá-la em detrimento de uma aplicação com um ambiente gráfico convencional. É também notório que a concentração das crianças e empenho em alcançar os objectivos é bastante superior, quando utilizam uma interface tangível.

¹Tecnologias da Informação e Comunicação

6.2 Trabalho futuro

Apesar de o sistema ter sido uma extensão a um trabalho já realizado, continua a haver lugar para mais melhoramentos. Relativamente à *Framework*, era útil existir uma aplicação que fizesse toda a assemblagem dos sensores. Ou seja, uma aplicação que conseguisse criar os módulos a serem utilizados pela *Framework*, por exemplo, através de uma ferramenta de autoria. Deste modo, seria permitido que todos os utilizadores, mesmo sem conhecimentos específicos de sistemas computacionais, pudessem, eles próprios, desenvolver estes módulos e estender as capacidades das aplicações que podem ser desenvolvidas através da *Framework*.

Seria interessante alterar a *Authoring Tool*, dando suporte a interfaces tangíveis, ou seja as aplicações serem criadas a partir de objectos tangíveis, ou através de tecnologia *multitouch*. Desta forma, poder-se-ia efectuar o estudo da implementação e criação com as interfaces tangíveis. Podendo também ser comparado com as interfaces gráficas. Para a *Authoring Tool*, era também importante acrescentar que nesta versão apenas é possível associar a cada estado um sensor. Desta forma, o estado torna-se muito mais simples, mas permite a criação de aplicações cujo grafo seja muito complexo. Desta forma, era bom desenvolver-se um estado que representasse uma aplicação constituída por estados mais simples. Deste modo, seria criado um nível de abstracção maior, a ponto de simplificar o grafo e o entendimento da aplicação por parte do utilizador, construindo assim uma representação hierárquica dos estados.

Deveria, também, ser estudado o efeito na aprendizagem a longo prazo, verificando qual das interfaces se mostraria mais apta em determinados contextos. Assim como verificar este estudo com crianças com incapacidades psico-motoras, visto nenhum dos utilizadores possuir qualquer tipo de incapacidade. Para além das crianças também para pessoas idosas ou com um nível de alfabetização mais reduzido, verificando qual das interfaces proporciona uma abordagem mais concisa na aprendizagem de alguns assuntos.

O estudo dos efeitos persuasivos na alteração de comportamentos requer, também, uma análise mais prolongada do uso do sistema, de forma a poder concluir-se como a utilização destas tecnologias pode influenciar os utilizadores a alterarem os seus hábitos.

A . Apêndice A, Manual de utilizador de *Authoring Tool*

Para se produzir aplicações interactivas, com recurso a interfaces tangíveis, na *Framework* desenvolvida nesta tese, é necessário configurá-las recorrendo a uma *Authoring Tool*. Assim este manual tem como objectivo auxiliar os utilizadores a conseguirem manipular a ferramenta para poderem desenvolver as aplicações persuasivas que desejarem.

Em primeira instância é necessário considerar os dois elementos que compõe a *Authoring Tool* (*Editor* e *Wizard*). Cada elemento tem objectivos diferentes e deste modo é necessário separar as explicações do funcionamento dos mesmos.

A.1 *Editor*

O editor é constituído por duas componentes um menu e um editor gráfico. No editor gráfico podem manipular-se os estados e as variáveis para que o grafo de interacção da aplicação seja visualmente fácil de compreender. Na figura A.1, encontra-se o exemplo de um grafo onde são visíveis os estados (Início e Fim), a ligação entre eles, mostrando que um acontece após o outro e por fim uma variável (teste). Na figura A.2, é possível verificar todos os menus e submenus existentes na *Authoring Tool*.

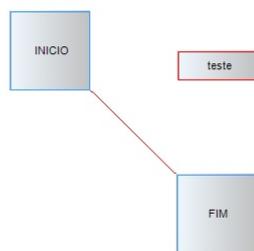


Figura A.1 Exemplo de um grafo

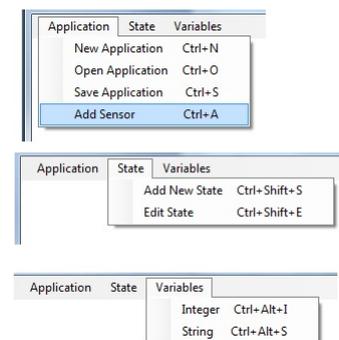


Figura A.2 Menu e submenus da *Authoring Tool*

- **Application**

- *New Application*, cria uma nova aplicação;
- *Open Application*, abre uma aplicação previamente guardada;

- *Save Application*, guarda, de forma persistente, uma aplicação;
- *Add Sensor*, adiciona um sensor à *Framework*.

- **State**

- *Add New State*, cria e adiciona um estado ao grafo (aplicação). Inicializa também o *wizard* (secção A.2), para a configuração do estado;
- *Edit State*, abre o *wizard*(secção A.2), para que o estado possa ser novamente configurado.

- **Variables** neste menu apenas são criadas e adicionadas variáveis ao grafo.

- *Integer*, variável do tipo inteiro (Numérico) ;
- *String*, variável do tipo *String* (Alfanumérico).

A.2 Wizard

O *Wizard*, permite a configuração das aplicações para que estas possam ser devidamente executadas pela *Framework*. Assim, quando um estado é criado o *wizard* é inicializado. Para se obter uma configuração correcta é necessário seguir os seguintes passos:

1. Atribuir um nome ao estado, este nome deve ser indicativo da acção que o estado efectua (figura A.3). Passar para o passo número dois (2).

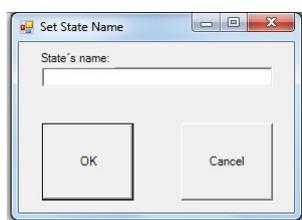


Figura A.3 Atribuir um nome ao estado



Figura A.4 Início de *wizard*

2. Após a atribuição do nome, pode ser escolhido iniciar a configuração (clicando em *start*) ou terminar a construção do estado (clicando em *cancel*), tal como é visível na figura A.4. Passar então para o passo três (3).

3. A partir deste passo já se inicia a verdadeira configuração do estado. Assim é permitido seleccionar uma acção intrínseca a cada estado, como reproduzir um vídeo, um som, alterar o fundo ou não executar qualquer acção. Este último é útil para configurar um estado que não altera o ambiente gráfico da aplicação, mas que altera propriedades nos sensores ou nas variáveis. (Figura A.5). Passa-se então para o passo quatro (4).
4. Este passo permite, definir a acção interna do sistema com a utilização dos sensores. Assim, como visto na figura A.6, pode-se escolher:
- *Send Input*(passo cinco (5));
 - *Get Output*(passo cinco (5));
 - *Variables* (passo onze (11));
 - *Nothing* (passo sete (7));

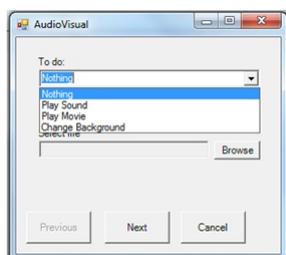


Figura A.5 Atribuir acção pré-definida ao estado

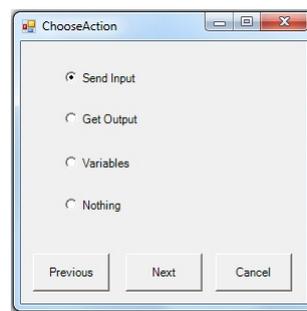


Figura A.6 Seleccionar a acção interna

5. Neste passo é permitido seleccionar o sensor e o respectivo método, que vão ser utilizados para o envio ou recepção de dados (figura A.7). Após esta selecção, o próximo passo depende do passo anterior a este. Caso tenha sido escolhido *Send input* (passar para o passo seis (6)). Caso tenha sido escolhido o *Get Output* (passar para o passo oito (8)).
6. Neste passo é permitido seleccionar os *inputs*, atribuir-lhes valores, provenientes de uma variável, como demonstrado na primeira linha (figura A.8) ou escrito explicitamente pelo utilizador, como na segunda linha. Após a configuração passa-se para o passo sete (7).
7. Este passo apenas define o próximo estado, que a *Framework* deve executar. Caso seja seleccionado *END*, será indicado à *Framework* que chegou ao fim de toda a aplicação. (figura A.9)



Figura A.7 Seleccionar um sensor e respectivo método

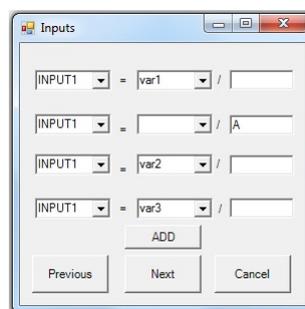


Figura A.8 Configurar *inputs*

8. Este passo permite definir o que fazer com os *outputs* provenientes dos sensores (figura A.10). Assim pode-se seleccionar entre guardá-los em variáveis (segundo para o passo nove (9)), ou manipulá-los directamente (segundo para o passo dez (10)).



Figura A.9 Seleccionar próximo estado a ser executado

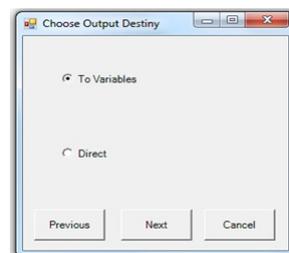


Figura A.10 Seleccionar destino dos *outputs*

9. Neste passo é permitido seleccionar as variáveis e afectá-las com o valor dos *outputs* provenientes do sensor (figura A.11). Passando de seguida para o passo sete (7).
10. Neste passo é permitido efectuar comparações com os *outputs* provenientes dos sensores, de notar que os dados são comparados e de seguida descartados pela *Framework*, sendo que para cada comparação é atribuído uma passagem para outro estado. (figura A.12)
11. Neste passo é permitido analisar os valores das variáveis declaradas no grafo da aplicação. De notar que para cada comparação está associada uma passagem para outro estado. (figura A.13)

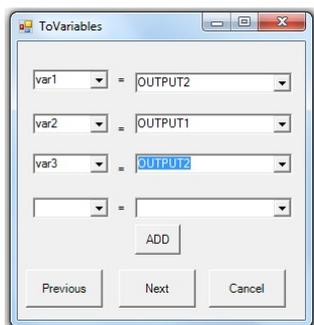


Figura A.11 Afectar valores de *outputs* às variáveis

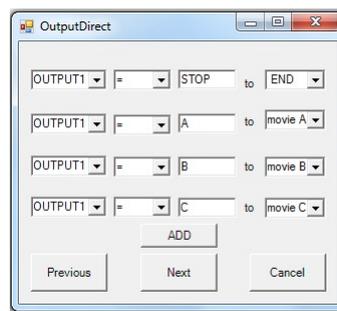


Figura A.12 Configuração directa dos *outputs* provenientes dos sensores

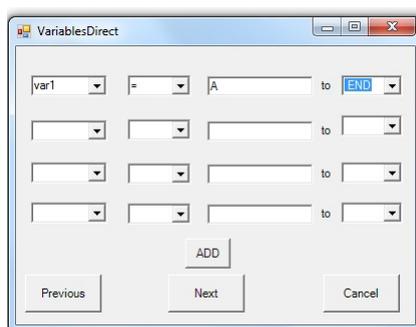


Figura A.13 Comparação dos valores das variáveis

Bibliografia

- [1] ANTLE, A. N., DROUMEVA, M., AND HA, D. Hands on what?: comparing children's mouse-based and tangible-based interaction. In *IDC '09: Proceedings of the 8th International Conference on Interaction Design and Children* (New York, NY, USA, 2009), ACM, pp. 80–88.
- [2] ARTOOLKIT. Artoolkit home page, 2009. "<http://www.hitl.washington.edu/artoolkit/>", acedido Junho 2009.
- [3] CHI, P.-Y. P., CHEN, J.-H., CHU, H.-H., AND LO, J.-L. Enabling calorie-aware cooking in a smart kitchen. In *PERSUASIVE '08: Proceedings of the 3rd international conference on Persuasive Technology* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 116–127.
- [4] FISCH, S. M. Making educational computer games "educational". In *IDC '05: Proceedings of the 2005 conference on Interaction design and children* (New York, NY, USA, 2005), ACM, pp. 56–61.
- [5] FOGG, B. J. *Persuasive Technology: Using Computers to Change What We Think and Do (The Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann, January 2003.
- [6] FOGG, B. J., AND IIZAWA, D. Online persuasion in facebook and mixi: A cross-cultural comparison. In *PERSUASIVE '08: Proceedings of the 3rd international conference on Persuasive Technology* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 35–46.
- [7] GIL-LEIVA, I., AND ALONSO-ARROYO, A. Keywords given by authors of scientific articles in database descriptors. *J. Am. Soc. Inf. Sci. Technol.* 58, 8 (2007), 1175–1187.
- [8] HARJUMAA, M., AND OINAS-KUKKONEN, H. Persuasion theories and it design. In *PERSUASIVE* (2007), pp. 311–314.
- [9] INTILLE, S. S. Ubiquitous computing technology for just-in-time motivation of behavior change. In *of Behavior Change, UbiHealth Workshop* (2003), vol. 2004.
- [10] ISHII, H., AND ULLMER, B. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1997), ACM, pp. 234–241.

- [11] JORDÀ, S., GEIGER, G., ALONSO, M., AND KALTENBRUNNER, M. The reactable: Exploring the synergy between live music performance and tabletop tangible interfaces. In *Proc. Intl. Conf. Tangible and Embedded Interaction (TEI07)* (2007).
- [12] KRANZ, M., SCHMIDT, A., MALDONADO, A., RUSU, R. B., BEETZ, M., HöRNLER, B., AND RIGOLL, G. Context-aware kitchen utilities. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction* (New York, NY, USA, 2007), ACM Press, pp. 213–214.
- [13] KROL, L. R., ALIAKSEYEU, D., AND SUBRAMANIAN, S. Haptic feedback in remote pointing. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems* (New York, NY, USA, 2009), ACM, pp. 3763–3768.
- [14] LOBO, P., ROMÃO, T., DIAS, A. E., AND DANADO, J. C. A framework to develop persuasive smart environments. In *Ambient Intelligence - Proceeding of AMI 2009* (2009), Springer-Verlag, pp. 225–234.
- [15] LOBO, P. J. G. Developing persuasive interfaces. FCT-UNL (2009).
- [16] MAHER, M. L., AND KIM, M. J. Do tangible user interfaces impact spatial cognition in collaborative design? In *CDVE* (2005), pp. 30–41.
- [17] MARAT RAFIKOV, A. B. W. Dinâmica e controle de regimes caóticos do sistema de lotka-volterra. UNIJUI.
- [18] MARKOPOULOS, P., READ, J., HOÿSNIEMI, J., AND MACFARLANE, S. Child computer interaction: advances in methodological research: Introduction to the special issue of cognition technology and work. *Cogn. Technol. Work* 10, 2 (2008), 79–81.
- [19] NORMAN, D. A. *The Design of Everyday Things*. Basic Books, September 2002.
- [20] PARMAR, V., KEYSON, D., AND DEBONT, C. Persuasive technology for shaping social beliefs of rural women in india: An approach based on the theory of planned behaviour. In *PERSUASIVE '08: Proceedings of the 3rd international conference on Persuasive Technology* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 104–115.

- [21] PATTEN, J., ISHII, H., HINES, J., AND PANGARO, G. Sensetable: a wireless object tracking platform for tangible user interfaces. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2001), ACM, pp. 253–260.
- [22] READ, J. C., AND MACFARLANE, S. Using the fun toolkit and other survey methods to gather opinions in child computer interaction. In *IDC '06: Proceedings of the 2006 conference on Interaction design and children* (New York, NY, USA, 2006), ACM, pp. 81–88.
- [23] ROMÃO, T., HIPÓLITO, J., CORREIA, N., CÂMARA, A., AND DANADO, J. C. Interactive public ambient displays for environmental persuasion. Poster in the 2nd International Conference on Persuasive Technology (Persuasive 2007), Stanford University, Palo Alto, CA, USA., 04 2007.
- [24] SPORKA, A. J., NĚMEC, V., AND SLAVÍK, P. Tangible newspaper for the visually impaired users. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2005), ACM, pp. 1809–1812.
- [25] TAKEUCHI, Y. *Global dynamical properties of Lotka-Volterra systems*. 1996.
- [26] TERRENGHI, L., KRANZ, M., HOLLEIS, P., AND SCHMIDT, A. A cube to learn: a tangible user interface for the design of a learning appliance. *Personal Ubiquitous Comput.* 10, 2-3 (2006), 153–158.
- [27] ULLMER, B., AND ISHII, H. Emerging frameworks for tangible user interfaces, 2001.
- [28] WEISER, M. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3, 3 (1999), 3–11.
- [29] WOOD, G., AND NEWBOROUGH, M. Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design. *Energy and Buildings* (September 2003), 821–841.