Preface

Lin Chao
Editor
Intel Technology Journal

This issue of the *Intel Technology Journal* is focused on fun—specifically on fun toys that use the personal computer. Today's children are very technologically savvy, and the computer plays a big part in their daily lives; therefore, "toys to think with and have fun with" make a lot of sense.

In 1998, Intel, together with Mattel Inc., designed and developed a new line of PC-enhanced toys, called Intel Play smart toys, that allow children to explore the world around them in a whole new way. These toys are low-cost consumer products designed to enhance the "magic" of toys for children by using the extended PC. The first product was the Intel® Play™ QX3™ Computer Microscope, which was unveiled at the 1999 American International Toy Fair. It was considered one of the most innovative new product introductions at the toy fair. With the microscope, children can magnify and display microscopic objects on their PC screens and then play with the images in creative ways. The microscope uses digital video-imaging technology to let kids view, enlarge, and save images of bugs, plants etc. The next year, 2000, three more Intel Play smart toys were introduced. These new PC-enhanced toys include the Computer Sound Morpher, the Digital Movie Creator, and the Me2Cam* Virtual Game System.

The six papers in this issue of the Q4, 2001 *Intel Technology Journal* provide excellent insights into the development of PC-enhanced low-cost toys. As well as presenting technical details of individual toys, a history of the Smart Toy Lab is also provided.

Why Toys?

By Herman D'Hooge
Manager, Design Center
Connected Products Division, Intel Corp.

In the first decade of the history of personal computing, processor performance could barely keep up with the demands of PC software and the evolution of even the most common desktop applications. This symbiotic interdependency was referred to as the "software spiral" and it fueled the need for faster microprocessors: new and better software applications demand faster processing, and faster processing, in turn, enables new and faster software applications. 3D games, multimedia, and real-time communication applications are all examples of the spiral model in action. Much of the focus of the software spiral effort has been on applications that lived within the confines of the "personal computer" desktop interaction paradigm. The personal computer paradigm is characterized and limited by requiring a person (the user) to sit at a desk, literally in front of the computer, hands on the keyboard and mouse and eyes on the computer screen.

Because of these limitations, the software spiral model had to be extended to include crops of new hardware devices or gadgets that offer user interaction to computing away from the desktop PC. This revised spiral model is now widely known as the Extended PC. This vision for expanding the PC's reach in the home is what will fuel demand for faster microprocessors during the next decade or so. It will set in motion an evolution that can dramatically change everyone's notion of what home computing is all about.

The Smart Toy Lab story and the products it has delivered illustrate the new uses (playthings) and new users (children) of the extended PC, and show how play experiences can reach their full potential with the computer.

It is often difficult to trace certain events back to their root cause. Intel's entrance into the PC-enhanced toy market is no exception. As far back as 1997, Intel Developer Relations, a part of the Intel Content Group, was active in directing the toy industry's attention towards technology and the PC. The concept and the economics made sense: by tapping into the interactivity and power of the PC, low-cost additions could be developed that—in combination with the PC—delivered a very powerful experience. "Transform a $99 toy into a $2000 experience" was the tagline in late 1997 (of course, reflecting today's drop in average price for a PC, this has become closer to a $1000 experience, but the concept is still sound).

A few years prior to 1997, applications research at the Intel® Architecture Labs in the context of its Anywhere in the Home computing capability initiative was exploring the reach and relevance of the PC in the home. Several prototypes of devices and appliances, including toys, as interfaces to computing, were developed and demonstrated.

Intel, from the corporate branding perspective, had a desire to learn from true consumer companies and extend the brand to reach real people. Major toy companies realized that low-cost electronics and software technologies opened up major new product opportunities for them and that tapping into the interactivity offered by the PC could help them recapture some of the older kids whose toys were replaced at an ever younger age by fast action video games.

The papers in this issue of the Intel Technology Journal provide some excellent insights into the development of PC-enhanced low-cost toys. The first paper describes the history of the Smart Toy Lab from its relatively independent and low-key formation as a joint project with Mattel Inc., through the birth of the Intel Play brand, to its organizational integration into Intel's Connected Products Division (CPD) and what the Smart Toy Lab is up to today as an independent business segment. This paper provides a contextual backdrop for the other technical and development process papers that highlight a diverse range of aspects of consumer product development. The paper on developing smart toys describes the typical process by which the majority of the Intel Play products have been developed and produced.

The Intel® Play™ QX3™ Computer Microscope was the very first Intel Play product to reach the market in the fall of 1999. It is an excellent example of a $99 add-on product providing a much more valuable experience. The QX3 Computer Microscope paper takes a look at some of the technologies that went into developing the QX3 and the tradeoffs that were made throughout its definition and development.

The Intel® Play™ Me2Cam* Virtual Game System is the first known commercial product that uses state-of-the-art computer vision technology including foreground-background segmentation. The paper on the Me2Cam does not explore the underlying technology, but rather, it looks at the design considerations that went into the definition of the application and play pattern that use the computer vision technology.

Fun and creativity with sounds is what the Computer Sound Morpher is about. This paper explores some key technology aspects of developing a product that is centered on the use of audio-processing algorithms.

The Digital Movie Creator paper explores the implementation of this product and the tradeoffs made during its development.

While the Smart Toy Lab is actively participating in the smart toy market segment, a parallel effort is underway to enable the toy industry as a whole. The goal of this enabling effort is to identify and remove hurdles that stand in the way of the "PC-enhanced toy" industry as a whole and that are best tackled through common industry solutions (i.e., standards). An example is the need for common low-cost protocols for two-way Toy-PC communication. A secondary, equally important goal of the enabling effort is that of industry education on technology and PC roadmaps and relationship brokering.

I hope you enjoy this feature issue of the Intel Technology Journal.

History of the Smart Toy Lab and Intel® Play™ Toys

Herman D'Hooge, Design Center (Connected Products Division), Intel Corp.
Melanie Goldstein, Smart Toy Lab (Connected Products Division), Intel Corp.

Index words: Consumer Products, Extended PC, Smart Toys, Intel® Play™ Toys

ABSTRACT

In 1998, Intel Corp. and Mattel, Inc. joined forces to create a "Smart Toy Lab" in a small, trendy office space located in the northwestern part of Portland, Oregon, USA. The lab was to bring together the best toy design and consumer marketing practices from Mattel with the technology expertise and innovation of Intel engineers. The opening of this office marked the beginning of the Intel Play brand of technology toys, starting with the flagship QX3™ Computer Microscope.

This paper reviews how and why this unlikely pair of collaborators came together, how the collaboration of resources and ideas worked over time, why the two companies parted ways, and why Intel has continued the development of Intel Play-branded products to date.

INTRODUCTION

In 1997, Intel was seeking to expand its reach into the consumer marketplace, and in particular into the children's market. They engaged in discussions with the toy industry giants and specifically with the market segment leader, Mattel, Inc. to better understand Mattel's efforts in the emerging field of interactive toys. Simultaneously, Mattel was exploring the idea of novel, interactive high-tech toy concepts that involved the use of the personal computer. Small sound chips and other low-cost, standalone technologies had been part of the toy industry for decades, but the quality of the electronically enhanced experience left a lot to be desired. Also, Mattel had a successful "interactive" division that developed children's CD-ROMs and game console products based upon their popular brands, such as Barbie* and Hot Wheels*. But Mattel was eager to do more. "Children are getting older younger," or CAGOY was the catch phrase, meaning that children stopped playing with toys at a younger age than a decade before. The traditional toy industry was significantly damaged by the emergence of the video game console marketplace; traditional toy companies entered this marketplace as secondary players, well after the market leaders who developed both the consoles and the platforms for game development. Mattel, and other major toy companies such as Lego and Hasbro, were determined not to be left behind in the next wave of technology to enter the children's marketplace; they wanted to recapture the video game enthusiast.

From these vantage points, Intel and Mattel made a decision to join forces to answer this fundamental question: what novel products emerge when you put Mattel toy designers together with Intel technologists and engineers?

SPRING 1998: FORMATION OF THE SMART TOY LAB

Executives from the Strategic Planning Department at Mattel and the Developer Relations Department (part of the Content Group) at Intel devised a business plan. The basic idea was a simple one: create an office space that merges an engineering team from Intel with a toy design team from Mattel. The team's mission was to invent and develop innovative products that would be the next "new thing" in toys. The lab would be located in the Portland, Oregon area to leverage the technology expertise, research, and technologies from the Intel® Architecture Labs in Hillsboro, Oregon. Executives from both companies were adamant that the office be located off campus–geographically separate from any main Mattel or Intel facilities. They wanted to empower this team to think and act as a startup and to liberate their operation from the standard operating procedures and slow decision making of either corporation. And thus the Intel/Mattel Smart Toy Lab (STL) was formed in a small office space

---

above an antique mall in a trendy shopping district of Northwest Portland.

The core team was intentionally kept very small, focusing exclusively on the invention, design, and development of the toys that were to hit the market in time for next year's holiday season. They drew upon the resources and expertise of the parent companies for many of their support functions as needed. This allowed the STL to select the best from each parent company. Furthermore, both companies learned from the strengths of their partner. Mattel, at its headquarters in El Segundo, California owned the marketing and sales of the products and the overseas manufacturing. Intel, at the Jones Farm campus in Hillsboro, Oregon and its headquarters in Santa Clara, California owned the technology and engineering related to the product's development and integration.

The team at the STL included three tech-savvy toy designers, one from Mattel's preschool division, one from Hasbro's boys division, and the third one from the independent toy inventor community. Four playful engineers were handpicked from the Intel Architecture Labs for their creativity, versatility, and innovative development experience. Finally, three producers were hired by Mattel to manage and coordinate the product development. The executive teams from the Intel Developer Relations Group and the Mattel Strategic Relations Group became the directors of the STL.

THE BIRTH OF THE INTEL PLAY BRAND

The original vision of the kinds of technology toys that ought to emerge from the intersection of Intel (technology leader) and Mattel (toy leader) came from joint brainstorming sessions. This vision remained largely unchanged during the first three years of the Smart Toy Lab (STL) and became the foundation of what later became the Intel Play brand.

The ground rules for a worthy Intel/Mattel toy were as follows:

1. **Fun**. Fun is synonymous with toys. Unless a toy delivers well on fun, nothing else matters. There currently is no IEEE standard that provides an objective and scientific measure for "fun," but focus testing usually provides a good idea.
2. **Open-ended**. Play patterns range from fully structured to fully open-ended. A structured play pattern consists of a fixed set of rules that bounds play. Television watching is an extreme example. Someone decides what it is you're going to watch and play just happens to the player. Video games provide a lot more interaction, but the rules and boundaries of the activity are clearly defined; the script is predefined by the game designer. Open-ended play has no rules. The child defines the rules and the play allows infinite variation. A ball is an excellent example of open-ended play. There are no rules that restrict ball play. Play with a friend, multiple friends, add a stick to play a baseball-type game, put pins on the floor for a bowling-type game, throw it, kick it, bounce it, make up your own rules that go far beyond what the toy designer may have imagined.
3. **Child is in control**. The child controls the pace of the play. Teddy bears stuffed with voice chips or other pieces of technology usually end up performing for the child. Those are examples of technology automating the play. This is the typical result when adding technology to an existing toy. We chose to always put the child in control and make the toy a tool in the hands of the child, its use only limited by the child's imagination.
4. **Challenging and creative**. Children seek instant gratification, but are also easily bored: "that's all I can do with a toy." If an activity is too difficult, it will become frustrating; if it is too easy the child quickly loses interest. Play that is challenging invites repeated use and is seen as providing more value to those who pay for the toy.
5. **Educational**. While playing with an educational toy is fun, the learning comes for free. This fact is not lost on parents who will often go to great lengths to direct their children towards toys that teach them something.
6. **Grows with the child**. As children grow older, they can continue to play with the same toy but in different ways. The child discovers, masters, and enjoys different features of the same product. This enhances the play value of the toy and often justifies the somewhat higher price of a good technology toy.
7. **Involve the Personal Computer**. Mention "Intel" to consumers and they immediately think Personal Computers. For the STL, this means that the PC plays an essential role in the toy's play pattern.
8. **Perceived to be high technology**. The goal for low-risk development for a nine months to one-year development cycle is to stay with well-understood and mature technology ingredients. However, it is important that the consumers, particularly the parents, perceive the toy to be high-tech. Daily interaction between engineers and toy designers helped the team marry the innovative industrial design and user interfaces with technology.

9. **Innovative**. As industry leaders, we want our products to be the first-of-a-kind, never seen before in a toy.
10. **At least one truly magical feature**. We learned this from Mattel and it still resonates strongly. A toy has to have that one special feature that makes a kid go "wow."

These ten golden rules were internalized by every individual on the development team and became the promise associated with Intel's toy brand. Intel brand strategists were engaged to help fine-tune, name, embody, and communicate this brand promise to parents and children. The Intel Play brand extension (or sub-brand) was created so that the Intel name would provide these toys with a mark of high quality and advanced technology. The Mattel logo would remain on the box to offer parents the assurance that this product would have great play value. The Intel Play sub-brand had its own logo, packaging style, and marketing materials, which appealed to children. This is in contrast to Intel's standard packaging and branding guidelines, which were not formulated with children in mind.

SUMMER 1998: CONCEPT CREATION

As the branding and messaging for the Intel Play line was being developed, the Smart Toy Lab (STL) development team rushed to create innovative product ideas that rated highly when measured against the ten ground rules. The first product ideas, along with the Intel Play strategy, were presented to Jill Barad, CEO of Mattel, and Andy Grove, Chairman of Intel, and to the Intel branding team for approval.

During the concept creation, Intel engineers got their initial exposure to the notion of *play pattern*, the specific ways a child uses a toy. Joint brainstorms thus far had only generated long lists of product ideas. To go from an idea to a fully developed play pattern is a long and involved process. This is where the Mattel toy designers applied their unique skill and magic. Simultaneously, the toy designers were introduced to a wealth of new state-of-the-art technologies from the Intel® Architecture Labs, as well as given insight into roadmaps for upcoming technology innovations.

It was made clear early on that the fundamentals of children's play are not defined by a specific toy or technology. These fundamentals have remained unchanged for as long as mankind has been around, and we quickly realized that not even the most advanced [1]technology was going to change that. Fundamental play values include exploration, discovery, creation, expression, imagination, nurturing, and collection.

However, technology, if used wisely, does provide new tools to the toy designer to define new play experiences, but they will always connect back to the fundamental play values.

Understanding that poor results were to be expected if technology was merely added to existing toy concepts, the team took a different approach. They set out to create entirely novel and innovative toy concepts that were uniquely enabled by technology and the PC. In other words, if the technology was taken away, the concept wouldn't be able to exist.

In the summer of 1998, eight concepts emerged that were considered to have reasonable potential and that warranted further exploration. It was felt that these concepts were worth presenting to the executive team as the initial crop and output of the companies collaborative experiment. These eight concepts were as follows:

1. **Internet Discovery Set**: This was a Radio Frequency (RF) tag reader that allowed children to navigate to Web sites using RF tags embedded into small physical toys. In other words, each of these toys would have a unique URL embedded within it. Placing the toy on the reader would point the PC's browser at the corresponding Web site (encoded as the RF identifier), and the child could engage in online activities related to the toy.
2. **Ultramind Magic 8-Ball**: This was a stress-sensing device that transferred biofeedback data to the computer and translated it into a series of "magic" and "fortune teller," or "truth or dare" games.
3. **Robox**: This was a PC/portable, artificial life game that allows children to build and groom robotic players for competition on their PC, then transfer them to a portable game device so that they could compete with their friends in the schoolyard.
4. **See Ya Bubba** (later renamed the Me2Cam): This was a series of immersive arcade-style games that utilize advanced computer-vision technology to transport a moving image of the child into game play.
5. **Computer Microscope**: This was a microscope with a PC-camera embedded as a replacement for the eyepiece. Users could view the magnified images on their PC screen in full color.
6. **Music Jammer**: This was a PC-connected musical instrument that allows children to create music with their PC by physically manipulating an abstract, tactile form. The user selects a specific instrument to control with the Jammer. The music is then

---

Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

automatically harmonized by the computer software, which also shows a cool visual representation of the music.

7. **PC Fun Phone**: This was a kid's pretend cell phone that connects to the PC via Radio Frequency (RF) transmission. The PC software would provide personal messages, jokes, and other features programmed at the PC and sent to the portable phone.
8. **PC Explorer**: This was a classic radio-controlled car equipped with a wireless video camera navigated and programmed from the PC.

Play patterns, expressed as story boards and product concept sketches, and potential industrial designs were developed by the toy designers while engineers developed product architectures, built working prototypes to validate their assumptions, experimented with solution alternatives, and tried to understand product costs. All eight concepts were tested in focus groups with children (the users) and parents (the purchasers).

FALL 1998: CONCEPT SELECTION

From the combined design, engineering, and market feasibility insights, three product concepts emerged–or perhaps more accurately, survived–as serious candidates for full productization and were given the green light. The selected concepts were the Intel® Play™ QX3™ Computer Microscope, the Internet Discovery Set, and the Intel® Play™ Me2Cam* Computer Video Camera.

All three products began development in earnest, but the Computer Microscope, given its perfect fit with the Intel Play brand and the fact that it had fully developed and rich and open-ended play patterns, quickly became the forerunner and flagship product.

Rather than growing a large internal organization with skills that might only be needed for the development of one product, external experts, developers, vendors, and suppliers were engaged to collaborate with the Smart Toy Lab (STL) on the development of the product.

The default model was to outsource development wherever possible, yet keep the overall program management and a limited amount of engineering work. By assigning STL engineers to key areas that linked the work of other vendors together, we kept our finger on the pulse of the project at all times.

Intel Play, Me2Cam, and QX3 are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

The STL staff delivered the industrial design, the product architecture, the middleware video streaming software, and the overall product integration and validation for the Computer Microscope. About ten external companies and/or internal groups within Intel or Mattel collaborated with the STL staff in the areas of optics, illumination, plastics, mechanical engineering, tooling, electronics and firmware, device drivers, application software, electronic registration, diagnostics, user-interface graphics and audio assets, packaging, and documentation. All these diverse areas were orchestrated by our producers/program managers.

The Internet Discovery Set and Me2Cam followed a similar development model. In early November 1998, after being presented with realistic cost estimates and a working prototype to communicate the concept, a focus group of parents unanimously decided that, although they loved the Internet Discovery Set concept, the price was simply too high for the perceived value. Our own insights into how complex it would be to develop this product to the full, considering the need for secure Web servers, custom kid-friendly browsers, and kid- appropriate and frequently changing Web sites, made for a quick decision. That was the end of the Internet Discovery Set. In the years to follow, more toy concepts were abandoned due more to product cost than to anything else.

And then there were two. The Computer Microscope and the Me2Cam.

FEBRUARY 1999: THE UNVEILING AT THE NEW YORK TOY FAIR

The International New York Toy Fair is where toy buyers meet toy manufacturers–where demand meets supply. Unlike major computer or electronics' tradeshows, the Toy Fair is not open to the public: rather, store chains that sell toys to consumers are guided through by appointment, and they provide initial estimates on how many units they expect to buy of a given product.

For the Intel Play line–and yes, two products do constitute a line–this was the first chance to see how the toy buyers would react to the products. Early working prototypes housed in plastic models of the real industrial design were demonstrated for two weeks straight. The reaction of the buyers was unanimous and overwhelmingly positive to both the $99(USD) QX3™ Computer Microscope and the $69(USD) Me2Cam* Virtual Game System–the full names assigned to these products.

Intel, Intel Play, Me2Cam, and QX3 are registered trademark or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

What until then was treated as an experiment by both parent companies suddenly became serious business with a long road ahead to get products fully developed, into mass production, and launched. The rush to get both products out the door in the remaining six months was on.

SPRING AND SUMMER 1999: SHIPPING THE QX3 AND THE ME2CAM

The QX3 was ready in September 1999, the Me2Cam in October. It was a long summer for everyone at the Smart Toy Lab (STL).

During the summer, the STL was adopted into the Connected Products Division, which at the time had only the Intel® PC Camera and Create & Share product line. Full organizational integration was to happen gradually.

Meanwhile, a whole new development cycle, starting with brainstorming for new ideas for year 2000 products, was already underway. Again, the team was forced into invention on a schedule, but this time they had the added load of products under full development.

The QX3™ Computer Microscope and the Me2Cam have each won numerous prestigious product awards in a range of categories. The QX3 was rated the top-selling multimedia toy of the 1999 holiday season despite a higher price than most other products in this category.

Winter 1999 and Spring 2000: Intel and Mattel–Changing Companies

In late Fall 1999, business conditions started shifting for Mattel and for the toy industry as a whole. Mattel wanted to focus on its core: traditional low-tech toys with strong kid appeal brands. The Toy Lab had also lost its champions within Mattel during this shift, and the parent company's visions about the future of smart toys and the STL started to diverge widely. These differences eventually started to permeate all facets of the collaboration from the ideas about the kind of products to develop, to marketing and merchandising strategies, to how to go about growing the business, and so on. In May 2000, Intel and Mattel decided to go their separate ways and formally end the joint project.

Intel's Connected Products Division, being bullish on the potential of PC-enhanced toys, decided to continue the investment in the STL and take the Intel® Play™ toy line forward. Intel hired key personnel from the Mattel team to help maintain the momentum at the STL. Two toy designers and three producers joined the Intel team, along with a Mattel sales director, who had an excellent network within the toy and mass-market channels.

From the crop of new toy ideas and concepts created during 1999, two product candidates emerged as the Intel Play products for the holiday 2000 season.

1. **Computer Sound Morpher**. This is a $49(USD) take-anywhere toy that allows children to gather sounds and take them back to a sound-editing and creative effects studio on the PC.
2. **Digital Movie Creator**. This is a $99(USD) product that allows children to make their own movies. It comes complete with a portable audio/video camera and easy-to-use movie-editing software with tons of special effects.

The Digital Movie Creator became a casualty of the separation from Mattel. Trying to complete the development of this product while untangling the toy lab from Mattel was judged too risky. Instead, Intel-only STL focused on just the Computer Sound Morpher. Successfully launching this product would prove that the team could successfully develop the product without Mattel. Packaging, marketing, operations, and manufacturing responsibility were moved to their respective Connected Product Division functional organizations.

Fall 2000: Computer Sound Morpher

Intel completed the development of the Computer Sound Morpher in August of 2000. The breakup with Mattel left the newly independent toy group scrambling to pick up the marketing from Mattel. By the time new marketing experts were in place, it was too late in the year to have significant impact on the holiday 2000 sales season. However, with the three Intel Play products in the market: the QX3™ Computer Microscope, the Me2Cam* Virtual Game System, and the Computer Sound Morpher, Intel Play did start to look more and more like a true product line.

Year 2001: Digital Movie Creator

After three years of product development, it was time to look both backward and forward, and strategize where the STL should be headed. Where previously children aged 4 to 12 were considered to be potential target audiences for Intel Play products, the team then decided to focus on the 10 to 13 pre-teen crowd. These children are very familiar and comfortable using personal computers; in fact most of them have never known a world without personal computers. Ideal product concepts for this audience would need to be less about play as an activity in its own right and more about gear that fit naturally into the busy lifestyles of these young people.

Many of the concepts readied for 2001 did not fit this change in direction and were abandoned. The mothballed Digital Movie Creator from the year before, however, did fit perfectly. It was dusted off and significantly refeatured and revised to include advances in low-cost cameras and trends in industrial design. It is the new Intel Play product for the year 2001. The extra year allowed the STL to make it a significantly better product for the same $99(USD) suggested retail price.

Going Forward

Since its release, people from a diverse range of disciplines and hobbies have discovered the capability of the QX3 Computer Microscope and have adopted it as a useful tool. Examples include science, stamp collecting, coin collecting, NASA's zero gravity clean room, forensics labs, archaeology, micropaleontology, circuit board inspection, and ophthalmology instrument inspection. New uses for this microscope are brought to our attention almost weekly.

The education community has also started to embrace the potential of both the QX3 and the Digital Movie Creator. Educators do not see these products as computer literacy items; rather, they see them as highly valuable and very affordable tools for improved science and social studies teaching. Curriculum development, teacher training, marketing, and distribution programs are in place to address the educational market segment. Intel Play products also have been introduced in the European and Asian markets.

CONCLUSIONS

Over the last three years, the Smart Toy Lab (STL) has matured into a fully staffed business unit within the Connected Products Division. This year, the combined efforts of this team have produced coordinated sales, marketing and merchandising efforts to broaden awareness of the Intel Play products, with the new Digital Movie Creator as the 2001 flagship product. The success of these efforts will come to light in the 2001 holiday selling season. Meanwhile, the team continues its efforts to bring innovative new Intel products into the lives of children. As part of Intel's corporate strategy, the STL clearly adds products to the Connected Products Division's consistent with the Extended PC directive, but it has also contributed a wealth of external design, sales, marketing, and development process experience (BKMs) to Intel's continued efforts in the consumer marketplace.

ACKNOWLEDGMENTS

The authors thank Michael Bruck (Intel), Doug Glen (Mattel) and Mary Ann Norris (Mattel), and the original Intel/Mattel executive team, for their vision and energy in driving the formation of the Smart Toy Lab.

AUTHORS' BIOGRAPHIES

Herman D'Hooge manages the Design Center for Intel's Connected Products Division. Prior to that, he managed New Concept Development at the Smart Toy Lab, which he co-founded in early 1998. During his 20 years with Intel he has held various positions in research, development, product architecture, industry evangelism, and management in areas ranging from multi-processor computer architectures, PC system architecture, operating systems, computer security, fault tolerance, distributed systems, computer telephony integration, and new computing applications research. He holds an M.S. degree in Electrical Engineering and an M.S. degree in Computer Science both from the University of Ghent, Belgium. His e-mail is herman.d.dhooge@intel.com.

Melanie Goldstein is Senior Program Manager at the Smart Toy Lab. She has over 10 years of program management experience in the multimedia and software development industry and was one of the founding members of the toy lab in 1998. She holds a B.A. degree in Political Theory from UCLA. Her e-mail is melanie.b.goldstein@intel.com.

Developing Smart Toys–From Idea to Product

Scott Boss, Smart Toy Lab (Connected Products Division), Intel Corp.
Henry Bruce, Smart Toy Lab (Connected Products Division), Intel Corp.
Charlie Case, Smart Toy Lab (Connected Products Division), Intel Corp.
Kendal Miller, Smart Toy Lab (Connected Products Division), Intel Corp.

Index words: Consumer Products, Extended PC, Smart Toys, Vendor Management, Vendor Qualification, Vendor Selection, Development Process, Product Life Cycle, Outsourcing, Product Integration

ABSTRACT

Creating an Intel® Play™ toy involves the collaboration of several different parties with a wide range of abilities. From toy designers, to engineers, to marketing operations and sales, the Intel Play team combines several disciplines to develop a creative play experience centered on the PC. This paper presents the life cycle of Intel Play toys by outlining and detailing the process used to create an end-user product from a budding idea. Starting with focus testing and concept approval, through product requirements and vendor selection, and finishing with architecture definition and product implementation, a lot of effort is expended before an Intel Play toy hits the shelf.

INTRODUCTION

Intel® Play™ toys are a combination of hardware and software that together form a unique play experience for children. The development of a Smart Toy involves parallel software and hardware efforts that must be continuously integrated to show progress and to validate architecture decisions. Software and hardware development at the Smart Toy Lab is a combination of product requirements, architecture definition, vendor selection, component development, and vendor management. Third-party vendors are engaged heavily for toy development, and this paper discusses the motivation, reasoning, and methods behind this approach. Integration brings the two worlds of hardware and software together through validation of the complete product in order to guarantee a high-quality, end-user experience.

On the software side, third-party vendors and software packages are evaluated in order to provide solutions for software features. Several factors are weighed when choosing software vendors including technical background, user-interface design, cost, quality, and integration logistics. The final software solution combines external and internal resources to complete the software picture.

On the hardware side, a similar evaluation process takes place with a heavy emphasis on component cost due to the low-price expectations of the toy industry. A wide range of skills is required from the vendor including the ability to provide cost-effective components, the ability to design a proper enclosure for the toy, the ability to develop firmware and device drivers, and the ability to provide a quality manufacturing process. A successful hardware design is the correct balance between hardware architecture and vendor selection.

The integration effort is a daunting task: software and hardware must be seamlessly integrated into a complete product. All functions of the hardware and software must be validated in all possible usage scenarios. The integration effort must simulate and validate consumer usage in the virtually limitless space of available consumer platforms and configurations (many hardware combinations, many operating systems, and even many languages if the product is to be sold in the international market). To increase the integration burden even further, the toy industry revolves around the Christmas season, which imposes a hard stop deadline by which the product must be finalized. The final quality of Intel Play products is determined by the integration effort, which ultimately sets the bar for the PC play experience.

Developing toys for the consumer marketplace presented Intel with a set of unique design, development, and integration challenges. These challenges are listed and described as well as the methods and approaches developed at the Smart Toy Lab for meeting these challenges.

The next time you walk into a toy store and see an Intel Play product on the shelf, think about the transformation that took place from a twinkle in a toy designer's eye to the shrink-wrapped package in front of your eyes.

TOY DEVELOPMENT OVERVIEW

Toy development at the Smart Toy Lab (STL) has evolved and has been refined quite a bit in the few short years that the toy lab has been around. Toy components, hardware vendors, and software vendors vary from product to product, but the overall method and approach to building a smart toy remains constant. Each toy can be broken down or dissected into high-level areas and components for both the hardware and software portions of the toy. The details of a typical toy are outlined in the Toy Anatomy section. The quality of the toy and the ease of development are tremendously impacted by the decisions made during the product definition and the product architecture phases. Even the best development efforts cannot completely overcome or cover up poor definition or architecture decisions.

The development stage of a toy must be rigorous and well-managed since there are several parties involved both inside and outside of Intel. The toy continually takes different shapes and moves in new directions throughout the development timeline. In many cases, several vendors make changes to hardware and software components in parallel. These changes must be tracked and validated with pre-defined project milestones to prevent unmanageable defects and to guarantee on-time delivery of the product.

CHALLENGES

Intel® Play™ toys present a set of challenges that are typically not found in one area within Intel. These challenges span several areas ranging from design (brain-storming, concept selection, user interface definition), through engineering (architecture, vendor selection and management), to marketing (young consumer audience, toy-selling season). Each area can dramatically influence the toy feature set so it is important to keep requirements from each of the areas in mind when defining the toy. With such a wide range of requirements, successful product implementation demands that all areas are well-represented and accounted for during product definition and development.

A challenge unique to toy design comes from the consumer expectation that toys should not cost a lot of money. Offering a low-cost toy is in itself an achievable goal; however, low cost quickly gets caught in a tug of war between high product quality and a rich product feature set. The Intel® brand name demands a high-quality product with a strong feature offering. While many toy manufacturers are able to compromise quality to save costs, Intel's premium brand name brings with it the highest expectations of quality. Toy designers and engineers must constantly evaluate the tradeoffs between cost, quality, and feature set when defining a smart toy. Evaluation of these tradeoffs must be thorough as it ultimately determines the product's success in the marketplace

TOY ANATOMY

Each Intel® Play™ toy can be summarized as a combination of physical hardware accompanied by a software suite (Figure 1). However, each toy varies tremendously at the lowest implementation level depending on its form and function. Basic design structures and typical components for hardware and software are outlined and described in this section.

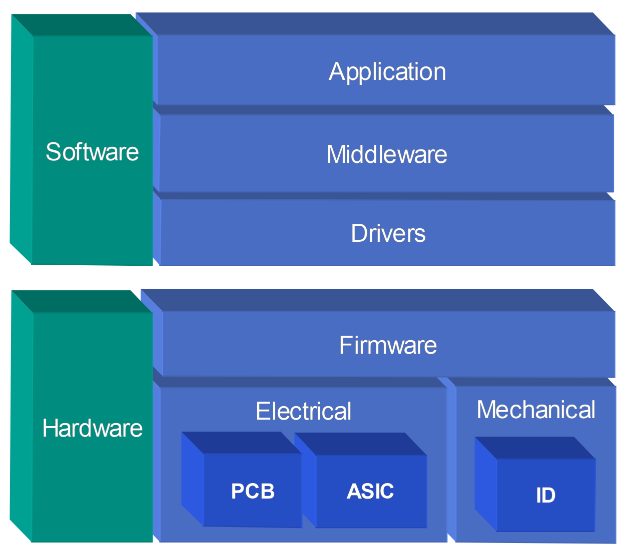

Figure 1: Toy anatomy

Hardware

The hardware of a smart toy, by its very nature, typically has an embedded microprocessor, sensors of one type or another, a mechanical system to comprehend moving parts, and some firmware to control and tie the pieces together. The hardware elements listed above can be categorized into three main components: firmware, electrical, and mechanical. The electrical and firmware engineers define the hardware architecture. The electrical engineer is responsible for the Printed Circuit Board (PCB) that forms the “heart” of the toy. The mechanical engineer works closely with the toy designer on the toy’s enclosure and ultimately determines how the PCB fits within the toy.

Firmware

Firmware is the software that runs on the micro-controller or Application Specific Integrated Circuit (ASIC), and it is more complex if the toy supports an untethered mode of operation. As an example, customized firmware was required for both the Computer Sound Morpher (CSM) and the Digital Movie Creator (DMC) since they both support capture of media away from the PC. The user interface portion of the firmware (i.e., “look and feel”) is specified by the toy designer, and implemented by the firmware engineer.

Firmware programming techniques and complexity can vary greatly between products. The CSM handled button presses and the related capture, playback, and deletion of audio clips. Due to the simple nature of the microcontroller architecture and its task, the firmware was coded as assembly language. The DMC firmware was more complex than the CSM firmware. For a start, there were two microcontrollers: one handled the user interface, and the other sequenced the camera’s audio and video processing. The user interface firmware was of equivalent complexity to the CSM firmware and was implemented in a similar nature. The audio and video processing firmware were more complex; they were based on the Intel® 8051 processor. The firmware was coded in C and cross-compiled to the target. In this case, the microcontroller’s performance was pushed to the limit to enable recording of both video and audio. The video was recorded at a frame rate of 10 frames per second with a frame size of 320x240 pixels. The audio was recorded at a rate of 12kHz (samples per second).

Electrical

The electrical components form the core of the toy’s operation. Usually, an Application Specific Integrated Circuit (ASIC) and/or microcontroller do the bulk of the work as this enables low hardware cost and a small form factor. Selection of these two components is a key stage of the toy’s development. If the toy can be used when not connected to the PC (often termed untethered operation), some form of status display (usually a customized liquid crystal display, or LCD) and memory are usually required. The electronics components are mounted on one or more PCBs to fit into the industrial design. Ideally just one PCB is used because inter-board connectors are expensive, and simple wiring can be unreliable.

Mechanical

Mechanical engineering (often termed M/E) is the “glue” that holds all the hardware together. Any moving parts required by the toy fall into the domain of the mechanical engineer and require thorough design up front. The mechanical engineer must also ensure that the toy is compliant with all toy safety regulations. This is a demanding task: tests must be conducted to ensure the product functions, with no sharp edges exposed, after being dropped and that there is no battery discharge if a small piece of wire is dropped into the toy’s enclosure.

A good mechanical design enables the industrial design to have the right external “look and feel,” the PCB to be neatly mounted inside the enclosure, and for the whole unit to be easily assembled.

Software

The software comprises three areas: drivers, middleware, and the application. Each area has its own purpose in the larger software whole and consists of a distinct functional set. The categorization of software into these areas provides for the following advantages:

- A large software project is broken into manageable pieces.
- There is a division of labor between in-house and out-of-house resources.
- A series of checks and balances is in place during the development cycle.

The details of the drivers, the middleware, and the application as well as the typical approach to their development are discussed next.

Divide and Conquer

Development of the software encompasses a wide range of activities, ranging from very low-level hardware communication programming to very high-level user-interface programming. Finding the necessary skill sets for the entire software stack in one place is often impossible. Therefore, we wrote the middleware in-house at the Smart Toy Lab and hired third parties for driver and application development. This approach is by no means the only solution, but it is now well-known at the Toy Lab and has proven to be successful for three separate products.

Drivers
Direct communication with the toy hardware is always a must for the software. Custom drivers are often necessary in order to implement toy-specific behavior. This driver work is always low level and is typically paired with the hardware provider. Interaction with the toy hardware ranges from simple input and output commands to full data streaming, depending on the toy. The toy feature set and toy-to-PC connections dictate heavily the amount and type of driver work that is needed. For example, the CSM required no driver work as it simply plugged into the already existing PC sound card and its driver. However, the QX3™ Computer Microscope and the DMC both required full streaming USB video driver solutions.

Middleware
Middleware serves several purposes including hardware abstraction, grouping of software features, and reduction of complexity. The main purpose of the middleware component is to abstract the hardware from the application by removing any direct communication with the driver from the application. The application uses the middleware components in order to interface with the hardware and to implement features beyond the scope of the application developer. Since the middleware is developed and validated at the Smart Toy Lab, it also serves the purpose of providing a vital porthole into both the application and driver development tasks. The communication between the application and driver can be monitored via the middleware and often application or driver defects can be short circuited and addressed in the middleware.

Application
The application provides all needed user-interface components as well as the majority of the functionality behind the user interface. It accesses the middleware components as necessary to implement complex features and to interface with the hardware. The entire multimedia application development process is encapsulated in this piece including media asset production and integration (art, sound, animations), component implementation and integration, and final application assembly. The look and feel of the software is carefully crafted here by experienced artists at a third-party development house with oversight and deep involvement by the toy designers at the Smart Toy Lab.

QX3 is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

TOY DEVELOPMENT

An Intel® Play™ toy starts as a new concept, moves through several approval stages during definition, matures during product development, and completes as a packaged end-user product ready for the store shelf.

Concepts and Prototypes

All toys start as a bright idea and then continue on through a myriad of refinements and approvals before there is any investment made to build a product. There are generally three types of prototypes used during the smart toy development life cycle.

During the early concept stage, product designers may make a variety of hard-foam or plastic models of potential products to help convey ideas, explore directions, and in some cases, test the concept with focus groups (Figure 2). While designers do some rapid foam prototyping in house, most of the models are made at outside shops by professional model makers, who employ Computer Numerically Controlled (CNC) machines and airbrush painting techniques to make photo-real, non-functional mockups.

Figure 2: Computer Sound Morpher plastic model

On the engineering side, often a concept breadboard is used to validate the fundamental technical risk in the project. For example, in the case of the Intel Play Computer Microscope toy, a prototype was built using an off-the-shelf optical microscope and a digital camera joined together (Figure 3). This combination was used to explore and specify magnifications, light levels, and project feasibility. The goal of the concept breadboard is to quickly understand the fundamental properties of the

Intel Play is a registered trademark or trademark of Intel Corporation or its subsidiaries in the United States and other countries.

toy, without necessarily worrying about size or cost at this stage.

Figure 3: QX3 microscope concept breadboard

The next stage of prototyping in the product life cycle is usually a form, fit, and function breadboard of the toy. This will generally consist of a Printed Circuit Board (PCB), machined or "rapid prototype" plastic parts, and software/firmware with very limited functionality (Figure 4). The purpose of this model is to validate the overall size of the product and provide an electronic breadboard for firmware development. This prototype presents a major challenge for the toy designer and hardware engineer, as they have to agree on an industrial design that satisfies both styling requirements and space envelope constraints. Often times, this prototype is also used at key marketing events to demonstrate the product.

Figure 4: QX3 final prototype

In addition to the several flavors of hardware prototypes, there are also software prototypes that are developed. Software prototypes are often simply push-button Windows* applications that demonstrate the ability to either implement a software feature from scratch or demonstrate the difficultly level of integrating a third-party software package. These kinds of prototypes are very important tools when determining the software feature list. A software prototype helps to crystallize the vision of the toy designer in some concrete form and helps the software engineer gauge the complexity of a software feature. By no means does the software prototype completely define a feature. It is simply a measuring stick that can be used as a piece of information when defining the software feature list, architecture, and user interface. It also serves as a resource load estimate tool when generating the project schedule.

Architecture Definition

In conjunction with the refinement and clarification of the toy concept, comes the definition and details on how to transform the concept into a product. Both hardware and software architectures are defined in parallel as it becomes clear exactly what features are required for the product. These two architectures are outlined in product architecture specifications, which contain component

---

*Other brands and names may be claimed as the property of others.

diagrams and sufficient implementation details on building the toy.

The software architecture must list implementation solutions for all required features for the software play pattern. The overall software architecture is fully described, and key technologies and potential vendor solutions are listed for each feature. This information becomes vital when constructing overall product schedules and cost estimates. It also serves to define functional development areas so that resources can be assigned accordingly. At this stage, the software architect must closely manage the delicate tradeoff between feature implementation and the software feature set. Decisions made during the architecture definition phase can have great impact during the product development phase and should not be taken lightly. Features must be analyzed and a clear development path must be identified for each feature before it can be included in the architecture.

The hardware architecture lists the major components to be used in the design; it tells how they must operate and how the toy interfaces with the user. The architecture also defines how the toy implementation is partitioned into an Application Specific Integrated Circuit (ASIC) and firmware and device driver components. The toy feature set typically demands leading-edge technology for implementation, so the architecture is developed in conjunction with the selection of a vendor that can deliver a leading-edge hardware solution.

In parallel with the hardware architecture, a mechanical design specification is drafted. The design specification covers all the dimensions of the product, which include how it fits together and how it looks.

Vendor Selection

External vendors play an important role in the development of Intel Play toys for both hardware and software development efforts. External vendors provide consulting expertise, development resources, and physical components for Intel Play toys. Vendor selection is an important step in the toy development process since the dependency on third-party vendors is a strong one.

Selecting a Software Vendor

The typical toy software is a full multimedia children's application with audio, animations, screen transitions, and lots of artwork. The workload for constructing the application can be divided into two areas, the user-interface development and the technology development that provides the functionality behind the user interface. Independent software vendors can provide solid solutions for toy software in both these areas. Identifying and selecting the best-qualified software vendor is a challenge in the toy development process. A discovery process is used to search for potential candidates. A small set of vendors is chosen based on their reputation within the industry and their relevant product experience. The chosen vendors are subsequently evaluated for their technical background and abilities, tools and process, cost, quality assurance, track record, and relevant software experience. A written questionnaire is used to gauge these criteria. Depending on how much the vendor will be involved with the product, a more rigorous evaluation is sometimes required. For example, application vendors are involved from the ground up: they contribute to the software framework, art assets, user interface, and the play pattern. As a result, the questionnaire for application vendors is intensive, and they are interviewed several times before a decision is made. However, individual software components are usually much less complex and often include existing product components that can be used "as-is" or "off-the- shelf." The evaluation of these vendors is, therefore, less rigorous.

Selecting a Hardware Vendor

Hardware vendor selection is a complex task, as there are many components to the hardware as presented in the following list.

- industrial design
- acoustic and/or optical design
- mechanical engineering
- electronic engineering
- firmware development
- electronic module build and test
- toy assembly and test
- packaging

One or more hardware vendors may need to be engaged based on the complexity of the toy. Ideally, a single vendor is selected to develop and manufacture all hardware components (electrical, mechanical, and firmware). Selection of a single hardware vendor is generally not possible because of the complexity of a technology toy. The hardware vendor selection process is broken into several phases. Initially, the focus is on finding a vendor that can provide the ASIC or heart of the hardware. Here, chip-set vendors that meet the requirements of the architecture definition are evaluated. If the selected chip-set vendor does not also provide firmware development services, then an additional vendor may be needed to fill this hole. If the firmware coding is very simple, the work is sometimes done in-house, but this doesn't usually happen.

The next step is to find a vendor for the development and manufacture of the electronic module. This vendor is often described as the manufacturing vendor. Development of the electronic module requires the design and layout of the PCB+. The electronic module then needs to be fitted into the toy enclosure. This is a specialist task requiring mechanical engineering design skills. Typically, the manufacturing vendor can both design and provide the molded plastic parts that form the "skin" of the toy. Finally, the toy needs to be placed in a package for the toy shelf. Intel has demanding standards in this area that not all vendors can meet, so packaging is mostly handled in house.

Selection of the manufacturing vendors is made in close consultation with the Intel operations team. The operations team must evaluate the vendor for stability and capacity capabilities. As there are sometimes partnerships between technology and production vendors, the latter may influence selection of the former.

Vendor Management

Simply selecting and hiring a third party to build components does not automatically produce a finished product. Engaging third parties to build components requires close monitoring and tight integration points for success.

The approach taken in vendor management must be flexible, depending on the core competencies of the vendor and the complexity of the project. The overall goal is always to deliver a quality product on time, on schedule, and at the right cost.

Several key tools are used to facilitate the development process. The first such tool is the project "Map Day," which occurs immediately after vendor selection. This is a mandatory on-site meeting wherein all stakeholders in the product development work out a full development schedule by negotiating key dependencies and integration points. The output of this meeting is the master project schedule that goes from project start to first customer shipment. It is some time after this period that the resulting integration timeline and its milestones (with relevant acceptance criteria) are baked into the vendor contracts.

With a schedule in place, the next step is to execute the agreed-upon plan. This is done via weekly team meetings between hardware and software vendors. These meetings are used to go over key issues and track progress. All parties will have team leads in their respective areas of expertise that will negotiate their way through development. Generally, these team leads serve as the sole contact points within their organizations for information exchange.

As Intel is the integration point for the hardware and software components, progress is also effectively tracked by validating interim deliveries of these components against the product specification and the Map Day schedule. Map Day is a meeting/process for planning new projects that require the integration of plans of multiple players or groups. As described below, integration points are designed to regularly validate the various components as they are integrated through the duration of the project.

Integration

Product integration attempts to provide the highest quality product for the customer, with minimal impact on the schedule, through constant evaluation of product health at defined checkpoints throughout the development cycle.

The Integration Engineer (IE) is the member of the core development team who is responsible for taking ownership of this particular challenge. Throughout the project, the IE remains focused on the final integrated product and its end user, while driving a variety of activities (called the integration process) that work to exhaustively validate the project schedule, product design, and product implementation.

Milestone Definition

If this integration effort is to be successful, it is critical to define a clear, rigorous, and methodical process. The starting point is a map of key milestones that mark the end of various phases: design and definition, pre-alpha, alpha, beta, release candidate, and finally a release to manufacturing. Using past products as a guideline, the durations between these milestones are scrutinized to ensure there is an appropriate amount of time to implement, test, and debug the features. Each milestone in this integration timeline contains exit criteria, each defined for that particular phase as an indicator for the status of the product. The IE will evaluate these criteria, and a resulting approval translates to a green light for various teams or external vendors to begin the next phase of the timeline. For those external parties that provide software or hardware deliverables, it also signals a contractual payment. However, if the product fails to meet the necessary criteria, those failures are evaluated and the team will do whatever is necessary (e.g., feature removal, additional engineering resources, etc.) to mitigate the risk to the next milestone and the overall project.

During the first phase, the IE is primarily architecting the appropriate validation. The engineer begins with an overall plan that describes the scope, the details of the timeline and milestone criteria, defect management, resources and tools, the various types of testing, a final approval checklist, and the methods and strategy required to successfully execute such an effort. Also during this planning phase, the IE contributes to the product definition effort. The engineer will provide input to the feasibility of the feature set and, eventually, the lower level details of functionality and requirements of an Intel Play product. This is where the IE has the opportunity to set expectations of product quality and also improve upon new products by rolling in support issues or customer feedback from existing products.

As the product definition and product architecture specifications become more concrete, a feature matrix is defined for the hardware and software components of the product. This matrix articulates what gets implemented and when this occurs. Such a matrix provides two major benefits to the project: clear expectations for each milestone, and a method to align the implementation of the various features and functions for each layer of the product. The latter will be constantly evaluated as the project progresses. Such complex products require this kind of regular validation; otherwise, the project is prone to miss critical milestones, as a result of delays stemming from misunderstood dependencies.

Test Methodology and Tools

The actual testing efforts are broken into basic acceptance, functional, interoperability, compatibility, field, and localization acceptance testing. The IE has to carefully analyze the schedule, project requirements, product specifications, and resources to properly coordinate and manage all of these pieces into an efficient and effective validation effort that sets the bar high enough to protect the good name of Intel.

The general approach remains a black box (or end user) testing. Each type of testing varies greatly in scope and in the resources required. The basic acceptance testing serves to provide quick feedback on stability and core functionality, with a general target of execution time running four to six hours or less. Functional acceptance testing encompasses the big portion of the overall test plan, as it validates all features of the product, both in terms of correct implementation and in the broader sense of product performance, stability, and behavior in stress/boundary conditions. Interoperability testing focuses on how well the product behaves with similar hardware or software products, or how well it works with certain operating system features. Compatibility testing attempts to validate the product on a sampling of computer platforms that are indicative of the kinds of computers being used by the target customers. Given the inherently complex nature of this task, the Intel Play team utilizes a selection process and as much market data as possible to design a matrix of computer systems that will expose as many potential issues with those customer platforms as possible, while remaining cost-effective in the number of systems and test hours required. Field-testing is a limited effort at acquiring early feedback from consumers with our product in their own homes or workplaces, using their own personal computers. This results in preliminary "real-world" data on potential usability and compatibility issues.

Certain tools are employed to track, utilize, or report all of these tests and their results. For the various test scripts, a proprietary test management system is used to create and maintain suites that contain the necessary procedures and their relevant scenarios. This is a database that contains a user interface that supports the authoring process, the assignment of procedures to a tester, and the generation of test-result reports. A third-party defect tracking system is used as a bug database (often one for software and one for hardware) for all internal and external team submissions and reports.

Not addressed in this document is component-level testing. Various software or firmware modules are always part of the overall integration effort, but the specific testing required for those pieces comes from a separate validation effort. The integration team coordinates with all of these teams, but the integration process always remains focused on the final product and its features from an end-user perspective.

Localization

Another factor to consider for an Intel Play toy is its readiness for other countries outside the United States. Intel Play currently delivers the English product version prior to any international versions, so the English product team does its due diligence to minimize the effort required on subsequent localization efforts. Therefore, some level of localization testing will occur on the English version. The goal of this effort is to ensure that the English version can run without any issues on international operating systems, the design is such that as few changes as possible will be required during the localization efforts, and that the localization kit included with the final gold release can be used by the localization team to easily modify assets and re-build the product for another language.

Product Implementation

As the product moves through the alpha and beta stages and becomes more stable and feature complete, the IE continues to increase validation efforts per the plan of record. The engineer works closely with other teams to communicate the results of testing and analysis against product requirements. In addition to overseeing a multitude of testing activities, the IE manages regular bug scrubs, drives ownership and prioritization of known issues, aids in the debugging effort through detailed characterization, and regularly reports product health to the various teams through indicators such as bug count, bug trends, and feature matrix status reports.

At the end of this integration process, the product enters the final two-week phase where it is evaluated against the golden checklist. Various procedures and checks are executed, and upon reaching the final seven days, all aspects of the release candidate are completely frozen. During the final seven days, the various tests continue until all validation procedures are verified to pass, known issues are resolved, all milestones are completed, and the product meets project requirements. The product is then declared golden for release to manufacturing.

Overall, the integration effort is a considerable task that, if orchestrated properly, will successfully serve to protect the brand name, maximize a positive customer experience, and minimize project risk.

SUMMARY

With a combination of hardware and software development and the integration savvy to merge the two, an Intel® Play™ toy evolves from a rough concept into a finished product. With considerable effort from all involved, the hardware transforms from breadboards and foam models to final circuit boards and highly polished plastics. The software follows a similar transformation from an artless engineering prototype to a full-blown graphical multimedia application. In the end, the final product on the shelf represents the full design, implementation, and integration effort of several vendors, all coordinated out of the Smart Toy Lab at Intel.

ACKNOWLEDGMENTS

Bob J. Hicke was the primary author of the original framework for the integration and vendor selection process described in this document. His vision has been expanded to create a strong foundation for the successful development of many Intel® Play™ toys.

AUTHORS' BIOGRAPHIES

Scott Boss is a software engineer who has been with Intel for eight years and has been working at the Smart Toy Lab for the past two years. Before the Toy Lab, Scott worked in the Intel® Architecture Labs on various multimedia projects ranging from audio infrastructures to 3D graphics. He received his B.S. degree in Computer Science from Valparaiso University in 1991 and his M.S. degree in Computer Science from Purdue University in 1993. His e-mail is scott.d.boss@intel.com.

Henry Bruce is a hardware engineer who has been with Intel at the Smart Toy Lab for a year. Before Intel, Henry worked for VLSI Vision Ltd. (VVL) in Edinburgh, latterly acquired by STMicroelectronics. As the software development manager he developed expertise in video processing and device drivers. He received his B.S. degree in Electronic Engineering in 1985 and his Ph.D. in Image Processing in 1993, both from the University of Edinburgh in Scotland. His e-mail is henry.bruce@intel.com.

Charlie Case has been a hardware engineer at the Intel Smart Toy Lab for two years. Prior to working at Intel, Charlie was Director of Engineering for ThrustMaster, Inc., a leading PC game controller company. And for eight years prior to that, he was product-engineering manager at Gyration, Inc., a wireless PC peripheral manufacturer. He received his B.S. degree in Mechanical Engineering from The University of California, Berkeley in 1985, and his M.S.M.E. degree from Stanford University in 1990. His e-mail is charlie.w.case.jr@intel.com.

Kendal Miller is an integration engineer who has been with Intel for five years and has been working at the Smart Toy Lab since it was formed three years ago. Prior to this, Kendal worked in Intel's OEM Platform Solutions Division working with several external vendors on various consumer software titles. He received his B.S. degree in Computer Science from the University of Louisiana at Lafayette in 1996. His e-mail is kendal.miller@intel.com.

Dissection of the Intel® Play™ QX3™ Computer Microscope

Lenka Jelinek, Smart Toy Lab (Connected Products Division), Intel Corp.
Geoff Peters, Smart Toy Lab (Connected Products Division), Intel Corp.
Jim Okuley, Smart Toy Lab (Connected Products Division), Intel Corp.
Steve McGowan, Advanced Engineering (Intel Architecture Labs), Intel Corp.

Index words: Consumer Products, Extended PC, Smart Toys, Intel® Play™, Computer Microscope, Imaging, Camera, USB, optics, CMOS

ABSTRACT

"Wow, it can do all that for $99 (USD)?" is the reaction of many people upon first seeing the Intel® Play™ QX3™ Computer Microscope in action.

Unlike regular optical microscopes, the QX3 has no eyepiece to look into. Instead it has a built-in camera that sends live video images of specimens or small objects at 10x, 60x, or 200x magnification to the PC via a Universal Serial Bus (USB) connection. The creativity software of the QX3 then allows scientists of all ages to easily view, capture, modify, and share images, videos, and time-lapse movies.

Unlike most commercially available microscope systems that offer on-screen viewing, the QX3 provides photomicrography at an affordable price along with additional functionality. Furthermore, the QX3 was designed for children; this translates into a device that is extremely easy for everyone to use.

So how did we do it? This paper examines the interworkings of the QX3 Computer Microscope, including both hardware and software aspects. It will become clear that a consumer product in the smart toy space is a complex, yet delicate balance between designing for "low cost" and remaining true to the vision of the product.

INTRODUCTION

A smart toy is defined as a plaything that uses technology in some preeminent way. Toys are the tools of play; they reduce the complex world of human culture to forms that children can grasp. So how do we create a smart toy that utilizes technology in a novel, ingenious way but yet focuses *not* on the technology itself, but instead on the *use* of this technology?

This is the challenge that we faced in the Smart Toy Lab in the spring of 1998. The result was the flagship of Intel® Play™, the QX3™ Computer Microscope.

The QX3 satisfies this challenge in these three ways:

- **Novel**: The major difference between the QX3 microscope and its predecessors is its ability to capture and view digital images and movies. Before the introduction of the QX3, photomicrography was only possible with the most expensive microscopes.
- **Savvy**: The QX3 microscope is easy to use; yet, it takes advantage of today's video imaging and computer technology.
- **Feasible**: The hardware design fits the cost constraints for a consumer product.

This paper discusses the tradeoffs and decisions behind making the QX3 not only a successful product for Intel Play, but also one of the key learning experiences in Intel's consumer product design.

Intel Play and QX3 are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

CONCEPT EVOLUTION

In many instances of product development, a new concept is a successful marriage of old and new, and sometimes even something blue. The QX3™ Computer Microscope is no exception. The concept of microscopy has been around for centuries; the first microscope is usually credited to Zacharias Jansen in the late 16th century.

Digital microscopy then utilized the power of the personal computer, and its professional applications in such fields as science and medicine are very well established. However, in the children's toy category, with a sub $100 price point, marrying the microscope with the PC was a unique challenge. A compelling, fun set of features that was acceptable to the target audience, children, was far from a trivial task.

To create a product with an open-ended play pattern was one of the main objectives of the project. An open-ended play pattern allows children to bring their own creativity to the various aspects of the product. Open-endedness not only prolongs the child's interest in the product, but also prolongs the life of the toy. Close-ended play patterns often result in toys that children get bored with and put in the closet. The goal was to create a long-lasting product that did not require the child to follow complicated instructions or rules.

A user-centered design approach was taken when developing the features of the QX3 microscope. Product concepts were generated that were easy enough for a six-year-old to figure out but involved enough so that older children would not get bored with them. This large set of product features was presented to target users in the form of models, storyboards, and software demos. Feedback from children in these tests was a major determining factor for the final feature set of the product.

The very first instantiation of the QX3 concept began as a sawed-off microscope with a digital camera attached to its eyepiece (Figure 1).

The first prototype provided significant learning: it was feasible to combine two off-the-shelf products and get a satisfactory outcome. This was also the first validation of cost implications. At this price point we could deliver this level of product quality.

Figure 1: First prototype

The next important validation came in the form of prototype number two: the hand-held digital microscope (Figure 2).

Figure 2: Hand-held prototype

In the design phase of the project, features were created that improved on the play of the traditional microscope toy. For example, a hand-held mode was developed in which the user could remove the barrel of the microscope from the stand to view specimens that would never fit on a traditional microscope stage, such as a flea on the family dog or the inside of a child's ear. Testing with the target users revealed that this was a very popular hands-on feature.

The design at this level also posed challenges: how do we fit the technology into a barrel that is safe yet small enough to fit into the hands of young children? How do we control the light conditions? And finally, would this feature be attractive to kids, not just engineers?

Building an actual prototype allowed us to validate the above questions to a level sufficient for further development. It also taught us another lesson: a prototype in hand is better than pages of engineering assumptions.

Function fitting the form became the reason for another milestone prototype (Figure 3).

Figure 3: Form and function prototype

Since the driving force behind the QX3 concept was not technology per se, but instead the use of the technology by our target audience. Its form in many ways dictated the technical function, at least at the implementation level. A key example was the layout of the Printed Circuit Board (PCB); the originally engineered design did not fit the form specified by the toy designer.

Safety was always an overriding goal of the product's hardware. Competitive microscope toys contain sharp instruments, glass slides, and even metal scalpels. Careful attention to detail was taken to avoid sharp edges on the plastic housing and also to design a product that could withstand heavy use and abuse by a child. Light bulbs on the QX3 are sealed safely behind plastic covers. Plastic specimen slides were found to be more than adequate in quality compared to the potentially more dangerous prepared-glass slides.

However, the evolution did not stop there. The next section describes the hardware and software features that made it into the final product as shown in Figure 4, along with the tough decisions that went into this difficult and final step.

Figure 4: QX3 Computer Microscope

TECHNOLOGY AGENDA

In the design of both the software and hardware of the Intel® Play™ QX3™ Computer Microscope, those features that could be accurately communicated to the end user via the product's packaging were specifically emphasized. Just because we can build it, doesn't mean we can sell it. This section elaborates on the technical tradeoffs made to create a successful $99USD toy. Included are the key engineering decisions made to support a viable feature set.

Given that the focus of all the features centers on one main feature, the microscope image on the PC, we needed to ensure that the product leverages the power of the PC and takes full advantage of the monitor for viewing

---

Intel Play and QX3 are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

images. Great care was taken in selecting the best camera sensor at the required low cost.

CMOS Versus CCD

The QX3 microscope uses a Complementary Metal Oxide Semiconductor (CMOS) sensor as opposed to a Charge Couple Device (CCD) sensor. The choice between the two options was a hot topic of discussion during the development cycle. With the entry of the Barbie* Digital Camera and the likes, the image sensor arena in consumer end products was in search of a cheap yet satisfactory solution. Everyone wanted to get the best quality for a minimum price. For the QX3, it eventually came down to cost: CMOS sensors are built upon standard CMOS processes; they have a key advantage of incorporating general-purpose digital logic into the sensor. CCD sensors, on the other hand, are strictly image sensors. The processes that are used for CCD sensors are poor at supporting random logic implementations, which means that the control electronics for CCD sensors are implemented as external circuits. CCDs also require multiple voltage levels, thereby complicating power-supply designs. Both of these issues raised part counts and increased costs.

CCDs do, however, have a wider dynamic range than CMOS sensors, thus they perform better in a variety of lighting conditions, especially in low light. Fortunately, the QX3 does not have to primarily deal with sunlight conditions. Low-light issues were resolved by carefully controlling the lighting environment of the QX3.

Lighting Environment

Power, safety, mechanical, illumination level, and reliability problems arose when illuminating the scene.

The first decision we had to make was regarding the source of the lighting: should we use LEDs or incandescent bulbs. LEDs do not burn out, thus they avoid the issue of providing replacement parts to end users. LEDs also are not made of glass, which circumvents potential safety issues regarding glass shards from a bulb that blew up or shattered from a drop. LEDs, however, suffered from poor luminosity: high-intensity LEDs may look bright, but they output less than 10% of the light from comparable halogen bulbs.

The final decision came down to cost. At the time, white LEDs were first coming onto the market. Our initial quote for white LEDs was $5; by the end of the project we were able to find white LEDs for $0.75 each. But, this was still much more than the $0.17 that we could pay for a miniature halogen bulb, similar to those used in small high-performance flashlights. So a decision was made to go with the halogen bulbs despite the design changes that would be required and the challenges that the halogen bulbs would represent.

Power

The halogen lamps that we selected are rated at 5 Volts, but the QX3 runs them at a slightly lower voltage to conserve bulb life and to lower the heat dissipation.

An early design decision was to rely solely on the Universal Serial Bus (USB) connection for all power. There are several advantages to this decision:

1. No need for batteries. This is important for a low cost of ownership from a consumer perspective.
2. No external power brick. Convenience and safety were key.
3. The user must only make a single connection to use the QX3, improving its ease of use.

As a result, the power for the lighting solution was tightly constrained to USB's power specifications. The halogen bulbs that we were looking at were rated at 350 milliAmps, which at first glance worked within our power budget. Unfortunately, an incandescent bulb looks like a dead short until it warms up. The solution was to design a current limit circuit that clamped the bulb current to 350mA; this caused the bulbs to warm up slowly (500 ms.) and prevented the initial current surge. USB power has strict constraints on current surge. Also, given that the maximum available power provided by USB is 500 mA, only one of the bulbs can be turned on at any given time.

Safety

The industry policy for toys is to enclose all glass light bulbs in a plastic cover to contain any shards if the bulb accidentally blows or shatters. The cover was problematic because it retained heat, causing the tungsten filament to evaporate, plating the inside of the bulb, and reducing light output over time. Moreover, if it got hot enough the plastic cover could melt. To minimize the possibility of excessive heating, the software turns off the lamp after a few minutes of no activity. We also attempted to increase the volume of the bulb cavity to allow more heat to dissipate. Finally, we took advantage of the fact that we needed a cover and made it translucent to better diffuse the light.

Mechanical

The mount for the incandescent bulbs needed to be removable so that it could be replaced. While this is not a significant engineering challenge, it is an instance when use of a LED-based light source would have circumvented a design issue. Both the upper and lower bulbs can be

*Other brands and names may be claimed as the property of others.

replaced with a small Phillips-head screwdriver. To ensure that the screws were not lost, we employed a captive design that allows them to be unscrewed without falling out of the cover assembly.

Microscopes generally use two different light sources depending on the specimen being observed: top for opaque objects and bottom for translucent objects. It was decided early to allow the QX3 to be removed from the base of the microscope. This feature permitted looking at the surfaces of large objects like ears and noses, versus what could fit on the microscope stage. One lamp was integrated into the microscope and the other into the base.

The top light is well-suited for illuminating opaque specimens such as coins and bugs. It is also a light source when the microscope is used in the hand-held mode, i.e., when the child removes the QX3 body from the base or the cradle as it is also referred to.

Controls were added into the software to allow the user to select either the top or the bottom lamp when the body was in the base. We also added a contact to the base that allowed software to determine when the body was in the base. The software would then switch automatically to the top lamp when the scope was removed from the base.

Illumination

A "Super Bright" white LED generates 0.014 Candle Power (CP) at 20mA, while a halogen bulb generates 0.06CP at 350mA. While a LED is more efficient, it takes many LEDs to generate the same amount of light as a single halogen bulb. Even though it was brighter, the light generated by the halogen bulb was still marginal at high magnifications. The field of view is inversely proportional to the magnification. The diagonals of the field of view are roughly 1.25", 0.227", and 0.0625" for 10x, 60x, and 200x, respectively. If we distributed the 0.06CP evenly over the large 1.25" diagonal field of view required by the 10x magnification, then there would be only about 5 percent of that light (0.003CP) that was hitting the small area displayed by the 200x field of view. We had to modify the camera firmware to increase the range of imager integration time to allow proper light collection under all magnifications.

Incandescent bulbs generate light that most people consider white, but as far as the image sensor is concerned it has a distinct red tint. This effect is referred to as the color temperature of the bulb. Firmware changes were required to force the color correction circuitry to its limit, in order to compensate for the color temperature of the light generated by the incandescent bulb.

Additional Lighting Controls

A time-lapse mode in the software supports capturing images at one-second to 60-minute intervals. In the longer time-lapse modes, the lights will turn off a few seconds after an image is recorded, and turn on approximately 15 seconds prior to taking a snapshot. The early turn on allows the color temperature of the lamp to settle prior to capturing an image, and the short turn-off time conserves bulb life during long time-lapse settings.

CIF Resolution–Striking the Right Balance

For the image sensor we used a CMOS Common Interchange Format (CIF) sensor. CIF sensors were developed for video teleconferencing applications. Their 352x288 resolution is a compromise between image quality and available bandwidth. Three years ago, during the development of the QX3, CIF was the sweet spot for low-cost USB video capture. The wide availability of CIF image sensors helped in obtaining lower pricing.

While the sensor provides CIF resolution, the camera hardware is set up to transmit a subset of the sensor array, 320x240 pixels to be exact, centered in the middle of the array. There were several advantages to using only the pixels in a central Region Of Interest (ROI).

With image sensors, there tends to be a slight degradation in the color quality of the pixels on the perimeter. Our image sensor has 352x288 physical pixels, and it places a Bayer-patterned mask over the individual pixels of the sensor to provide color information. A Bayer pattern covers 4 pixel groups with 1 blue, 1 red, and 2 green color masks, respectively. (The extra green pixel is used to compensate for the increased sensitivity of the human eye for green.) The color value of an individual pixel is computed by evaluating the intensity and color contributions of the adjacent pixels. Needless to say, the color calculations are compromised for pixels around the edge of the array, where there are no adjacent pixels. Pixels along the outer edge are also more susceptible to greater lens distortion. These problems are circumvented by not including the outer-edge pixels in our images.

Another beneficial side effect of using a smaller ROI is an increased frame rate, because there are less pixels per frame to send through the USB pipe.

IMAGE COMPRESSION

The CMOS sensor is integrated with a Digital Signal Processing (DSP) component. The DSP provides additional processing on the sensor's data, such as compression. In our case, the frame rate was limited by the bandwidth of a USB connection. The maximum throughput on a high-speed isochronous USB is limited to 1.023MB/sec. Most cameras use a lower rate to allow some bandwidth for other USB devices to work while the camera is streaming video.

There is always a tradeoff between image quality and frame rate when it comes to compression. The higher the compression setting, the more frames of data can be sent through the USB; therefore, the frame rate is higher. However, the video compression used by cameras is lossy, i.e., higher compression settings decrease the quality of the image. With no compression, the original quality of the image is preserved, but the frame rate is lower. After careful evaluation of the tradeoffs, we chose to use no compression of the image data for the QX3™ Computer Microscope. This resulted in an image quality that gave the end user the best picture detail. The resulting frame rate was approximately four frames per second, which is slow, but acceptable for the application.

Lenses and Magnification

The QX3 microscope has three preset magnification options: 10x, 60x and 200x. The user can change magnifications by manually rotating a barrel, which contains three lens tubes. Classically, microscopes have “objectives” that rotate into position at the bottom of a long lens tube and replaceable “eye pieces” at the top of the tube. This solution is flexible because it allows a wide variety of magnifications, but it is expensive because multiple lenses are required in each of the actual objectives to condition the light so that it can span the length of the lens tube without distortion. By using a lens tube for each magnification, we were able to optimally position the lenses in the tubes and minimize the lens count.

The “magnification” of the QX3 is the ratio of the field of view on the specimen stage to the size of the image on the monitor. The magnification is comprised of three components: optical, pixel scaling, and digital. The magnifications provided by the pixel scaling and digital components are fixed.

The *optical* magnification for 10x, 60x and 200x is performed via the custom lens system, where the lenses magnify the field of view on the specimen stage by 0.2x, 1.1x and 4x, respectively.

The *pixel scaling* is simply the ratio between the image sensor pixel size and the monitor pixel size. The pixels on a 15” monitor, running at 800x600 resolution are roughly 270x270 µm. When the 9x9 µm pixels of the image sensor are displayed on the monitor, the sensor pixels are “magnified” 30 times.

The *digital* magnification is performed when the 320x240 image is then software interpolated to the final 512x384 pixel resolution that is displayed on the monitor, resulting in an additional 1.6x magnification.

The advertised magnifications for the QX3 assume a 15” monitor. The actual magnification depends on the size of the monitor. In all cases, the QX3 application forces the video mode of the monitor to 800x600.

The optical magnification is the key to good performance in the QX3. The decision to perform 1.6x software interpolation was mostly for aesthetic reasons. On an 800x600 display, the resulting on-screen image was large enough for user viewing, and it left enough real estate around the Live View window for the application’s buttons and user-interface elements.

From a feature perspective, the 10x magnification was designed to provide as much depth of field as possible for easy focusing with the limited light source and simple optics. The field of view for the 10x magnification was also an important factor in its design. The field of view had to support a specific play feature that tested great with children: the ability to use the microscope to acquire small photographs of friends or family members’ faces, cut them out using the software, and then paste them onto the heads of bugs. Essentially, children wanted to also use the microscope as a scanner.

Also, as objects are increasingly magnified, they become less recognizable, especially to children. The 10x or “wide-angle” magnification allowed children to see giant versions of insects that they could recognize.

The 200x magnification was largely driven by a marketing requirement. 200x allowed the QX3 to compete with other entry-level laboratory microscopes in the marketplace.

Lens System

The lenses were designed for optimum quality, ease of operation, and safety, while remaining within the target range price of a toy.

One of the goals of our lens design was to allow changing magnification with minimal refocusing on the user’s part. Parafocal lenses do just that: they provide magnification changes without the need for manual refocusing. The QX3 microscope does not have parafocal lenses, but it was designed to get as close to that functionality as possible. The working distance of all three QX3 microscope objectives lies approximately between 26-29 millimeters, but again, the objectives are not parafocal with one another.

To achieve parafocal lenses in the QX3, we would need to add additional factory alignment (axially) and would have to make design and manufacturing changes so that the mechanical system would be robust enough to hold focus while changing objectives. This is not a trivial task. It should be noted that even high-end microscopes aren’t exactly parafocal at high magnifications.

Based on the tradeoff between cost, quality, and safety, a decision was made early in the design phase to use plastic lens elements instead of glass ones.

Another technology that the QX3 took advantage of was "binary" or "diffractive" optics. When light is refracted through a lens, some colors are bent more than others, causing a rainbow effect around the edges of objects. Typically, lenses are paired to correct this problem. The "binary" feature of the lenses in the QX3 etched virtually invisible concentric ridges on one surface of a lens. These ridges provided a diffraction grating, similar to a Fresnel lens, which acted as if we had placed a second lens in the optical path.

A key goal was to minimize the number of lens elements needed. This was accomplished through the use of our novel lens barrel approach and diffractive optics. The use of a single lens in the higher power objectives was specifically to reduce optical aberrations. The QX3 only uses four lenses, three of which are binary: one for 60x, one for 200x, and two for the 10x magnification.

Having non-glass lenses helped us achieve a stringent safety and reliability compliance. This is significant in areas such as drop testing, choke hazards, etc.

Finally, the lenses were specifically designed for manufacturing, with foolproof assembly techniques. The lenses themselves were manufactured in the United States for best quality control.

Focus Mechanism

The QX3 can be focused in one of two ways. In the cradle mode (where the QX3 body is held by its cradle), the focusing happens by manually moving the stage up and down via a focusing knob. In hand-held mode (where the user holds the QX3 body in the hand), focusing happens by moving the QX3 body closer and farther away from the object until the image is in focus. The hand-held mode is best suited for the 10x magnification setting. Higher magnifications are more difficult to use in this mode since focusing on the object is more sensitive to hand movements/shaking. Most users, especially children, have difficulty holding the QX3 body steady enough to provide a steady picture image. To make hand-held focusing easier, there is a "foot" on the bottom of the body that positions the optics at the proper working distance when the body is standing on a flat surface. Rocking the body on the foot can perform fine focusing adjustments.

The benefits of being able to use the QX3 microscope in the hand-held mode far outweighed the focusing difficulties, hence this design decision. The hand-held mode greatly expands the world that the child is able to examine at the microscopic level. With traditional microscopes, the user is limited to only examining specimens that fit on the microscope's stage. The QX3 microscope removed this barrier.

Exposure Control

While the user has the ability to control the lights in the QX3 microscope, direct exposure control is done by the application. The application adjusts the exposure and color balance levels to provide the optimum setting under various lighting conditions. For example, when the QX3 is removed from its cradle (i.e., is in hand-held mode) and the user turns off the light, the application switches to day light color settings and adjusts the gain levels.

SOFTWARE FEATURE METHODOLOGY

One underlying design mantra for the Intel® Play™ QX3™ Computer Microscope is to "harness the power of the PC to make a $99USD toy yield a $1000USD play experience." High-tech software running on the host computer best delivers on this mantra. In early discussions, the potential for heavily algorithmic features became overwhelmingly obvious. Potential features included image stitching and mosaicing, time-lapse microphotography, image processing to improve depth of field, heavy algorithmic image quality enhancements, and object tracking and identification, to name but a few. In these initial engineering-driven discussions, the following criteria, in order of priority, were used:

1. Do the engineers think it is cool?
2. Does it use the latest, cutting-edge technologies?
3. Can we build it?

While we as engineers clearly feel that this will deliver a superior product at the cost of all else, a high-tech product, however, does not imply a marketable product. First, the technology must be easily conveyable in order to sell the product, a common problem with leading-edge ideas. Second, the technology must work solidly: consumers have little tolerance for cool features that are inconsistent and partially usable. Lastly, the appearance of high-tech is often more successful than actual high-tech: low-tech solutions to a stunning feature are more stable and lower risk than high-tech solutions.

Using these observations, an interdisciplinary group of software engineers, toy designers, integration engineers, and marketing representatives revisited the software feature list. This group enforced a greater, more accurate litmus test for feature inclusion, a test that stressed the unique and targeted use of technology in the eyes of the end user over the complexity of the technology. Thus, the refined acceptance criteria for features were as follows:

1. How does the feature enable and potentially enhance the core feature of the product: the display of the microscopic image on the computer?
2. Is the target audience interested? A significant realization during development was that there was not a single person in the product development team that was a member of our core target audience. Similarly, will the target audience understand it? How hard is it to convey the feature?
3. Can it be implemented to gold quality in time?
4. Will it run on a typical user's computer (i.e., a minimum system configuration)?
5. How much validation effort will it take to ensure a gold-quality solution?
6. What is the "wow" factor? (Note: this does not necessarily imply a highly technical feature.)

This litmus test resulted in a fully functional application that includes: a) a live mode where the user can preview and record still images and movies; b) an image-editing mode complete with common painting tools; c) a "special effects" image filter mode; and d) an image and movie slide-show creator.

With the acceptance criteria enumerated above, we now will examine in more depth, five features and design decisions that were considered for the QX3 software package. Of these five elements, four were included in some form in the final product.

Image Stitching

The software development group for the QX3 included several engineers with image-processing backgrounds. This led to an initial emphasis on highly algorithmic image-processing-based features, including image stitching. The idea was to use the hand-held microscope (or a stationary scope with a motorized stage) to stitch together multiple highly magnified images into an extremely large mosaic super-image. This would potentially turn the microscope into an extremely high-resolution scanner.

Stepping through the engineering-driven criteria list, this feature hit the bull's eye. It was very cool, used the latest image-processing technologies, and could be implemented. For the refined criteria list, however, image stitching performed miserably. The concept is somewhat difficult to convey, and was deemed to be less important to the core 8-14 year-old target audience. Its cutting-edge technology would be difficult to implement to gold quality in minimal time and would substantially increase the validation effort. Lastly, the algorithm would likely not be usable on our minimum system configuration. As a result, this high-tech feature was not chosen for inclusion.

Time-Lapse Photography

Another featured considered was the ability to take time-lapse movies. Here, the computer takes a snapshot anywhere from every second to every 60 minutes and then compiles the images into a movie. This feature is a great example of a high-tech feature that can be implemented by comparatively low-tech solutions.

The largest risk associated with this feature is the significant validation effort required to ensure that the feature is bug free. First, the validation tests have to run over extremely long periods, tying up valuable testing resources. Secondly, it would be useful to allow the user to use other programs during a long time-lapse recording. This further increased interoperability testing.

In the end, this feature survived in the final product, but was modified to prevent the user from easily using other applications concurrently. This experience proved that validating the implementation of a feature is just as important as the actual implementation of the feature.

Printing

The ability to print from the application is often looked at as fundamentally necessary and entirely *un*interesting to the engineers. For our target age group, however, the ability to print is essential due to the target age group's desire for possessing tangible evidence of their work. Therefore, the printing capabilities were expanded to include the ability to print stickers and large posters, requiring some image interpolation algorithms to ensure image quality. In this instance, we have a low-tech feature that required significantly more substantial technology for success.

Image Filters

The QX3 software includes a set of filters that can be applied to an image, creating special-effect renderings. The filters include a kaleidoscope, bug's eyes transformations, and other warping image-processing algorithms. Here, each filter effect is a discreet action applied to an image or movie. As a result, validation and interoperability with other application features is minimized, allowing the developers to implement self-contained, high-tech features. As the QX3 has evolved, new optimized filters for the latest processors have been cleanly added as the product has been refreshed.

Middleware

Some of the heaviest lifting for an application often occurs in the "middleware," or the plumbing of the application. Again, software engineers are presented with the decision on how to architect the application, and how to integrate the application with the operating system. In the case of middleware, the question of high-tech now becomes "how much of the operating system's new features should we take advantage of?" This is especially problematic when comparing the engineers' criteria list with the refined product's criteria list. Here, we have to balance the new capabilities of the latest operating system features with the stability and wider end-user presence of older solutions.

At the risk of accepting a "not invented here" mentality, it may even be acceptable to implement certain new technologies found in the operating system in-house, to ensure that any last-minute bugs can be addressed internally before shipping. Fundamentally, application developers are fairly chained to any bugs pre-existing in the operating system.

For instance, Windows Driver Model* (WDM) streaming was a new technology hitting the mainstream with Microsoft Windows* 98. This infrastructure would have been ideal for use with the QX3. The QX3, however, was initially designed to additionally support the Microsoft Windows 95 platform, which at that time displayed problematic symptoms with WDM streaming. Therefore, we committed to an older VfW-based solution wrapped into the newer DirectShow* framework, part of Microsoft DirectX*, to ensure Windows 95 interoperability. Ironically, the Windows 95 requirement was later dropped after further testing. Subsequent releases of the QX3 are then expected to migrate fully to the WDM solution.

In summary, features and design architectures were best chosen due to a variety of concerns, only a few of which were engineering driven. Many times, high-tech solutions were not the optimal solution. Even the most optimal engineering solutions did not always prove to be optimal for the end result of the product as a whole. Equally distributing ownership feature resolution to a variety of well-respected disciplines resulted in a product that best fulfilled its promise.

FUTURE CHALLENGES

Keeping the technology fresh is an important premise for any smart toy concept. During the product design it was our intent to design scalable hardware and software to assist with future product redevelopments and refreshes.

The intended use of the Intel® Play™ QX3™ Computer Microscope was always as an end-consumer product. To our fascination, however, it went way beyond our 8-14 year-old target audience. For instance, the QX3 has had wide acceptance from stamp collectors to paleontologists, and from NASA engineers to surgical instrument companies. The potential for widespread use of computer-based photomicroscopy keeps expanding.

The original QX3 was developed three years ago. Given the pace of technological progress, the future opens up possibilities that were not feasible then: ROI digital zoom, better picture with VGA resolution, new optics, better live imaging effects and editing, better focusing, and higher magnification. Some risky image-processing algorithms have also stabilized, thus warranting inclusion.

As the power of the PC increases, so does the potential for the "magic" behind connected toys such as the QX3 microscope. The element of instant gratification is becoming a part of the culture. Three years ago it took a few seconds to apply a filter to an image for a special effect; now it happens almost as instantaneously as the child's finger hits the function button. Advances such as these will continue to influence smart toys of the future.

ACKNOWLEDGMENTS

The authors express their appreciation of Vareck Bostrom, Phil Doberenz, and Philip Lantz for their insightful feedback in reviewing this paper.

We also acknowledge Michael Davidson at Florida State University for his outstanding Web site on digital microscopy using the QX3, located at http://www.microscopy.fsu.edu/

AUTHORS' BIOGRAPHIES

Lenka Jelinek has been a "versatile engineer" with the Smart Toy Lab since its onset to its end. She was an engineering architect on the QX3 and with passion continues to ideate and innovate smart toys of the future. She holds degrees in Electrical Engineering and Computer Science from the University of Wisconsin at Madison. Her e-mail is Lenka.M.Jelinek@intel.com

Geoff Peters also was an engineering architect for the QX3, and has a master's degree in Computer Science from Purdue University. His professional interests include semantic video processing, robotics, advanced

*Other brands and names may be claimed as the property of others.

computer-human interface design, and any other discipline that yields depth of thought. His e-mail is Geoff.Peters@intel.com.

Jim Okuley has over 10 years experience designing and inventing for the toy industry. He holds a degree in Industrial Design from the University of Cincinnati. He has been an Intel Senior Industrial Designer at the Smart Toy Lab since it started. Jim is also a designer of other award-winning toys such as the Intel® Play™ Computer Sound Morpher and the recently released Intel® Play™ Digital Movie Creator. His e-mail is Jim.Okuley@intel.com.

Steve ("Maven") McGowan has a degree in Electrical Engineering from Marquette University. His previous experience in designing the optics and electronics for head-mounted display systems was invaluable during the development of the QX3. His e-mail is Steve.McGowan@intel.com.

Intel Play is a registered trademark or trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Game Design Principles for the Intel® Play™ Me2Cam* Virtual Game System

Herman D'Hooge, Connected Products Division, Intel Corp.
Michael Goldsmith, Intel Labs, Intel Corp.

Index words: Consumer Products, Extended PC, Smart Toys, Computer Vision, Machine Vision, Video as Input, Gesture Recognition, Motion Detection, Video Image Segmentation, Virtual Blue Screening, Natural Interfaces, Human Computer Interfaces, Head Tracking, Hand Tracking, Usability

ABSTRACT

Despite the long history of computer vision in academic and industry research circles, few real-world applications that use computer vision technology have actually been developed, let alone found their way into the homes of consumers. With the advent of fast processing, affordable and easy-to-install and use PC cameras, and advances in video-processing algorithms, the underlying technology has become sufficiently capable to enable certain classes of applications on average home PCs.

Natural interfaces have long been said to be the holy grail of computing that will revolutionize how people interact with computers. We've all heard claims that the keyboard and mouse will soon be a thing of the past, with people instead interacting with computing via speech and gesture. Whereas fulfilling these claims requires the accuracy of the technology to improve substantially, they usually ignore the more fundamental question of whether or not it is really desirable for people to use speech or gesture in the first place or under what circumstances it makes sense to do so.

This paper explores the *usage*–not the technology implementation–of computer vision in one commercial product developed jointly by Intel and Mattel. The product is the Intel® Play™ Me2Cam* Virtual Game System, designed for children aged four to eight. Informed by actual human behavior, the nature and limitations inherent in the technology have led the game designers to formulate a set of specific design rules that have guided the design of the application. This paper explores these design principles.

THE INTEL® PLAY™ ME2CAM* VIRTUAL GAME SYSTEM

In October 1999, Mattel launched the Intel® Play™ Me2Cam* Virtual Game System, one of the first, if not *the* first, commercially available PC application products targeted for consumers that was built using computer vision technologies. The retail product includes a USB-connected PC video camera (the Me2Cam) with Windows* device drivers and the Virtual Game software, which consists of five computer games and some related activities. Once the application is launched, the player's physical motions control all activities of the software as observed by the Me2Cam camera, as detected by the computer vision software technology, and as interpreted by the specific application.

Like all tethered PC cameras, the Me2Cam camera is typically placed permanently on the computer's monitor facing towards the person sitting, or in this case standing and moving, in front of the computer monitor where the game activity is observed.

Throughout all activities, the player sees a mirror video image of himself or herself immersed in a virtual cartoon-like graphical world. Interaction with this world is through moving one's body or by touching objects in this virtual environment with either hands or the head. The virtual world responds to the player's actions in intuitive and interesting ways.

The five games are modeled and reached as physical places. The player can move from place to place on *Main Street* (Figure 1) by leaning slightly off-center to the left or right. This has the effect of moving the player either left or right thereby passing various places, just as one passes houses while walking on a street. An activity is selected and initiated by the player standing still (moving one's body back to center) for a few seconds in front of the place representing the desired activity.

Figure 1: Main Street

In the *Bubble Mania* game (Figure 2), the player finds himself standing in front of a giant bubble-making machine. The machine shoots bubbles into the room that come down in a slow-moving swirl around the player's body. If the player touches a bubble with a hand, the bubble pops and the player scores points. The objective of the game is to score as many points as possible before time runs out. To make the basic game more interesting, a number of bubbles with special behaviors are introduced as the game progresses. Some of these have desirable behavior whereas others are better avoided. For example, if a player pops a bubble with a red cross on it, the time clock will be replenished. However, if a player pops a jail bubble, the player's video image will be taken and shrunk, so it fits within the bubble where it will remain for several seconds while the bubble floats around, thereby wasting valuable time. If a player pops a firecracker bubble, the player's video image is broken into multiple pieces and scattered across the screen, again wasting time. Finally, if a player hits a bubble with his or her head, the bubble will bounce instead of pop, thereby providing the player a mechanism to move trouble bubbles out of the way.

Figure 2: Bubble Mania game

The *Snow Surfin'* game (Figure 3) places the player on top of a mountain on a surfboard, sliding downhill. The slopes have obstacles such as trees, roaming animals such as penguins, bears, and raccoons, and also ski jumps. The player steers the snowboard by leaning left and right, avoiding the obstacles that reduce the player's speed and prevent the player from getting to the bottom of the mountain before time runs out.

Figure 3: Snow Surfin' game

Club Tune (Figure 4) is all about movement and dance. The speed and intensity of a virtual band of musicians on stage is controlled by the amount of movement generated by the player. Stand still and the band will get bored and go to sleep. The player can chose from among several bands and types of music; the player also has a drum pad on the left for adding special drum effects during the activity.

Figure 4: Club Tune Dance Hall

The *Fun Zone* (Figure 5) activity is not a game with a specific goal. It moves the player's image through a series of special effects, which far surpass those of the familiar fun house of mirrors found in carnivals, shrinking, stretching, morphing, and transforming the player's video image as it passes through the stations on the way.

Figure 5: Fun Zone activity

Finally, for a true aerobic workout, there is *Pinball* (Figure 6). The virtual world is an actual pinball machine. The player's video image is shrunk in size and replicated five times as the actual bumpers and flippers of the pinball machine. As with regular pinball, points are scored by keeping the ball bouncing off various elements of the machine as long as possible. In this computer game, this is accomplished by hitting the ball with any body part thereby simulating the physics of the moving pinball.

Figure 6: Pinball game

VIDEO AS INPUT TECHNOLOGIES

All activities in the product make use of a small number of computer vision technology components, sometimes referred to as *video as input* technologies. The Intel® Play™ Me2Cam* Computer Video Camera just captures and streams live video to the computer. This video stream contains everything that is in the field of view of the camera: the player's body, a section of a child's room with decorated walls, shelves with books and toys, etc.

To immerse the player's video image inside a virtual environment requires that the background be removed from the source video stream. The process or technology for determining which parts of a video stream are part of the foreground image and which are part of the background is referred to as *foreground-background segmentation*. For the purposes of the virtual game system, the foreground image is the player, and the background image is the image of the room–in other words, everything else.

The knowledge of which pixels in a video frame belong to the foreground and which belong to the background, then, allows the original video stream to be modified. This is done by removing all image elements that are considered to be part of the background and eventually substituting them with other images or graphics. This is often referred to as *virtual blue screening*. Instead of making the foreground/background determination based on a certain color in the background image (chroma keying), in the case of the Me2Cam, the determination must be made differently since the background can be arbitrary.

Intel Play and Me2Cam are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

In the *Bubble Mania* game, the behavior of a bubble is different depending on whether the bubble is touched by the player's head or a hand. The application must be able to determine the *x-y* location of the player's head and hands in order for it to be able to detect whether the head or the hand is touching the bubble; with this knowledge, it activates the appropriate behavior. A simple heuristic model of the human upper body is used by the computer vision *head-tracking* and *hand-tracking* algorithms to estimate the position of the player's head, left hand, and right hand.

In *Club Tune*, the amount of motion in the video stream is what drives the virtual world's behavior. The *motion-detection* algorithm provides a measure of the amount of motion in the video stream at a given point in time.

All applications in the Virtual Game System are built on these three technologies: (1) foreground-background segmentation, (2) head tracking and left/right hand tracking, and (3) motion detection.

This paper does not go into detailed descriptions of these algorithms or explore what makes one implementation superior. Instead, we focus on the design of the games and on the applications that use the technology.

DESIGNED FOR USE IN REAL HOMES

The Intel® Play™ Me2Cam* Computer Video Camera is designed to be used by *real* people in their *real* homes and in whatever location and lighting conditions exist around their computers. Requiring consumers to move things around in their homes too much in order to use this product isn't realistic.

From an application design viewpoint, this adds requirements dictated by the *context of use* of the product and has significant implications for the product's design and engineering.

With the camera placed on top of the computer monitor, the source video stream is whatever happens to be in the field of view of the camera. Personal computers are located in a diverse range of home offices, children's rooms, or in the corners of family rooms. The background for the camera images can be made up of virtually anything, from white walls, to cluttered shelves with books and toys, to open rooms. This is why foreground-background segmentation must perform robustly with virtually any type of background.

Intel Play and Me2Cam are registered trademarks or trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

The foreground-background segmentation includes an initialization phase each time the application is launched. During this phase, the player is asked to step out of the camera's field of view for a few seconds. This allows the software to record and register the background. After the initialization phase, the player can step back into the camera's field of view and play can begin. Segmentation will remove all pixels from any frame in the video stream that matches the registered background. What's left over is a video stream of just the foreground–in this case, the person playing the game.

To the segmentation algorithm, anything that does not match the registered background is considered foreground and displayed as if it is the person's video image; anything that does match the registered background is considered background and is removed. That means that anything in the image that changes over time relative to the background that was registered during initialization will be visible on screen. Flickering televisions or computer screens, other people or pets walking by, curtains waving in front of an open window, or clocks, will all become part of the foreground and may cause undesirable interference in the control of a specific game activity.

The set-up program of the Me2Cam helps the player in adjusting the play environment by identifying sources of motion in the background and suggesting actions for eliminating them and improving the game play experience.

Any sudden and dramatic change in the ambient room lighting on the background, such as someone turning a light on or off or clouds blocking/unblocking sunlight, will be perceived as different from the registered background scene and may cause the background to suddenly become visible. Also, bumping the camera, so that there is a slight shift in what is in the camera's field of view, can result in the whole camera image being interpreted as foreground.

These conditions do happen. If they happen frequently enough, they can easily become a nuisance and render the experience worthless. The application therefore has to deal with them and devise mechanisms to recover on the fly with minimal interruption to the actual game play. If foreground-background segmentation starts failing in the middle of game play (as observed by the player), he or she can step out of the camera's field of view for a few seconds. The computer vision recovery algorithm will then detect the lack of any motion and register the image as the new background from there on.

Some conditions are more difficult to design around. If the distance between the player and the background is short and lighting is frontal, harsh shadows of the player may be cast onto the background. These shadows will move as the player moves and appear as foreground.

Wearing a white shirt in front of a white wall may cause the player's image to match the registered background. If that happens, then the result is that the player's video image may have "holes" in the body where the background shows through. In the set-up program, players are advised to change clothing to increase foreground-background color contrast, which is more user friendly than requesting the player repaint the room.

When designing the Me2Cam, we envisioned localizing it for a number of countries and geographies outside of the US. Electrical power distribution in the majority of the target countries is either 50Hz or 60Hz (or both, like in parts of Japan). This is significant in that fluorescent light output tends to pulsate with the electrical power frequency. When captured with an imaging sensor, this causes horizontal banding in the image that may slowly scroll up or down the camera video image. Algorithms exist that eliminate this banding from the camera's image, but in order for these algorithms to work, the frequency needs to be known. Interestingly enough, there is no robust way to automatically determine the electrical frequency, so the user installing the product is being asked this obscure question. Again, this is an artifact of developing products for the real world.

Finally, studies done in people's actual homes have shown that the area around the computer in the home is dimly lit. When it's too dark, it will be difficult to make out details in the player's face. The set-up program again suggests adjustments to the room light level for an optimal play experience.

ABOUT PC CAMERAS

Most high-resolution, high-frame-rate PC video cameras, common in video conferencing or used as webcams, use compression to reduce the bandwidth demands for moving video bits into the computer. The video stream is then decompressed by software running on the host PC. Compression/decompression introduces both latency (buffering, time to compress, time to decompress) and compression artifacts that may make the decompressed image appear noisy.

In a game where your physical motion controls the game behavior and the player uses visual on-screen feedback (closing the feedback loop), latency must be kept to a minimum. Furthermore, any noise or artifacts in the image used for segmentation may show up as foreground motion, potentially making it difficult to segment out static background since everything may appear to be moving.

For these reasons, the Intel® Play™ Me2Cam* Computer Video Camera uses both low resolution (120x180 pixels) and no compression. It operates at a high enough frame rate and a low enough end-to-end latency to make the games playable and responsive.

Most video conferencing cameras provide automatic and continuous adjustment of the camera's controls such as overall image brightness and white balance. If the player moves in front of the camera, the overall light level/brightness of the image is affected. If the auto adjustments are on, the camera will try to adjust the settings, thereby affecting the background's brightness. The adjustment will cause the background over time to deviate from the registered background, and it will show through as foreground. Once the game starts, auto camera settings must, therefore, be disabled.

Even with technology and precautionary measures in place to handle these real-world conditions, imaging limitations and computer vision technology are far from perfect. The next section takes a look at how the application or play pattern was designed such that these known technology limitations do not become application weaknesses. This art is what is referred to as play patterns or application design, and it is the field of expertise of interaction designers and product designers or toy designers. These applications require one to take a fundamentally human-centered viewpoint of the experience that's being created as opposed to a technology-centered one.

CHILDREN AGED FOUR TO EIGHT

The activities included in the product have been designed for use by children aged four to eight who are standing up and moving in front of the camera. These are small humans with limited computer literacy skills. Sure, they may be able to use a computer mouse, but mostly they just want to play and have fun.

These children have little patience for anything that doesn't work or requires a long learning curve to get started. They want instant gratification, but at the same time don't want the games to quickly become boring. Most cannot read and would not understand error messages, even if reported verbally. Further, they need to be verbally prompted through the program startup.

Finally, these children have reflexes and behaviors that are different from those of an adult and that are, in many ways, much more straightforward and logical.

Game Design Principles

While the following principles may seem almost trivial, it is essential that the activity designer fully internalize what they are in order to design activities that deliver well within the constraints of the interaction medium. The rational/analysis is provided first, followed by a statement of the principles it supports.

The games are all controlled by the player's physical position or motion, and also by showing a real-time mirror video image on screen as captured by the camera and segmented by the foreground-background segmentation. For this to be effective at all times, the player has to stay within the field of view of the camera. A wider camera field of view picks up a broader scene that would allow the player to move a greater distance while still in the camera's field of view. However, a larger field of view does result in a proportionally smaller image of the player on the screen. This image could be digitally enlarged but that reduces the quality and the magic of recognizing one's personal features in the image.

While some left-right movement is possible, too much movement and the player strays outside of the camera's field of view or is too far removed from the screen to be able to see. That implies, for example, that "walking" past objects, as a metaphor for controlling on-screen scrolling, is not natural. The player just can't keep walking or very soon she'll be out of the picture. The player essentially has to remain in the same spot and merely lean left or right, or at best move a small step sideways in either direction. This leads to the first principle, P1.

P1: The Player's Left/Right Movement is Very Limited

Most computer monitors in the home are 15 inches or 17 inches in size. Being able to see oneself with enough detail on the screen from where one is standing is an essential aspect of this game. Moving too close to the camera will cause the player to be only able to see her forehead; moving too far from the screen will cause the player not to be able to see the screen or would cause the player to bump into a wall or furniture. Also, as the player moves forward or backward, the ratio of the player's image size on screen relative to the size of on-screen objects in the virtual world will change. This both destroys the designer's vision of the virtual world, and it also has the effect of having the player's image obscure areas of the virtual environment. Player motion is therefore to be confined to a plane at a fixed distance from the screen and camera. Hence, we have the second principle, P2.

P2: The Player's Distance From the Camera is Essentially Fixed

The camera's field of view is only able to capture the player's upper body, from the waist up. This restriction is due to the small size of the captured image. Capturing more of the player's body would result in the player's face being represented in too few pixels. That rules out the use of the player's legs or feet as a way to control the activities. It also implies that hands can be tracked only when they are not in front of the body.

The *Snow Surfin'* game, for example, places the player's upper body on a computer graphic bottom body that sits atop the snowboard to simulate a whole body. This also rules out games derived from sports such as soccer where foot action is key. This brings us to the third principle, P3.

P3: Only the Player's Upper Torso is in the Game

The player has to look directly at the computer screen to see what's going on. With the camera on top of the monitor, a player will always see herself looking forward. That has implications for the kinds of activities that make sense. Having a conversation with a cartoon character isn't natural if the character is depicted beside you.

For example, in the *Snow Surfin'* game, the player's on-screen representation is always moving towards the player. To be more exact, the player isn't really moving; instead the snow landscape around the player is scrolling. While it may seem more intuitive to be facing the other way, simulating a true first-person view for this game, children liked seeing their faces on screen and didn't mind at all that they were surfing in that direction. This brings us to the fourth principle, P4.

P4: The Player Always Faces Forward

The child will see his or her own image on screen. For many children the fact that they can see themselves on the screen in the game is a large portion of the appeal of the experience. The image acts as a mirror to the player since that is what all of us are intuitively expecting the behavior to be.

The activities are designed with a clear reason why the player sees himself in the game. This cannot be a keyboard and mouse replacement. If an activity is easier performed with a mouse, then using one's body motion instead can quickly become a frustrating experience. We, therefore, defined activities or worlds that fundamentally require a physical and mechanical behavior: i.e., the hand and body action are the natural way to interact. Popping falling bubbles, hitting a pinball, and leaning to the left or right to steer a snowboard are all physical behaviors. This leads to the following two related principles, P5 and P6.

P5: Body Motion and Vision Cannot be a Keyboard or Mouse Replacement

The state of the art in computer vision is approximately where speech recognition was a decade or more ago. The technology is not 100% robust. Attempts to replace the keyboard with speech, even with today's speech engines, are still largely unsatisfying, except in special cases. In both speech recognition and computer vision, there must be a compelling need for the input modality. In speech, this is frequently a hands-free/eyes-free requirement. For computer vision in games, the motivation is fun; and that fun is what allows the player to tolerate, and even enjoy computer vision.

P6: There Must be a Valid Reason for the Player's Image to be in the Game

The resolution or precision of a computer mouse or typical pointing device as operated by a typical player is about a few screen pixels squared. That makes it possible for mouse-controlled computer applications to have large numbers of controls packed on a single computer screen.

In contrast, the resolution achieved with hand or head interaction with objects in a virtual world is very low. The relatively large on-screen hand size, user's imprecision of hand orientation and movement, non-zero latency, and imprecision in the vision technology, all contribute to making this a low-bandwidth user interface. The minimum size of on-screen objects being controlled is roughly of the same order of magnitude as the hand controlling it. Thus, we have the seventh principle, P7.

P7: The User Interface Resolution is Very Low

In the design of the Intel® Play™ Me2Cam* Virtual Game System's activities, the resolution of the user interface implies that only a small number of distinct active controls can exist on any given screen and that they are spatially separated. For example, the floating bubbles are the active controls in the *Bubble Mania* game. Intersecting one's hand with the bubble activates a bubble's popping behavior. The hand is a more-or-less blob-shaped object with an approximate *x-y* center position, and a bubble is roughly the same. When both *x-y* positions are within some threshold, a collision occurs, activating the bubble.

Although popping bubbles for points in a child's game is far from mission critical, the illusion of the virtual world has to be sufficiently close to physical world bubble popping for this to be believable and not become frustrating.

It is a fact of human anatomy that our hands connect to our arms and our arms connect to our bodies at the shoulders. Human heads are fairly fixed to the middle and top of the human upper torso. The spatial range of where head, hands, and body can be moved in the virtual world on the computer screen is severely limited. Game design must take this into account when placing objects in the virtual world.

The most logical location of the objects that can be activated is in a semi circle around the player's body and above the head. They must be close enough so they can be reached; yet, they can't intersect with the body.

On the other hand, controls that may have undesired effects if activated at the wrong time should be harder to reach or activate. Exiting an activity requires standing motionless on the exit control for a few seconds– something that is unlikely to happen accidentally in an energetic game. This brings us to the eighth principle, P8.

P8: Common User Interface Objects Must be Easy to Reach; Infrequently Accessed Controls Must Require a More Deliberate Action to Activate

It has to be instantly clear to the player what is expected of her and how she should interact with the environment. While this is good user-interaction design in general, it is especially true in the case of a game for young children.

Game designers deliberately rejected the idea of using specific gestures for control. Gestures, no matter how basic, would have to be learned by the player and be such that they would be unlikely to occur during normal game play. That doesn't match with the need for instant gratification. Also, some game play can become pretty involved, as when trying to achieve a new high score, so accidentally activating the "exit" gesture command would be highly undesirable. And this brings us to the ninth principle, P9.

P9: Intuitive Interface; No Learned Gestures

Having each virtual world be based on a familiar physical world setting creates instant familiarity with what seems fun and logical activities within that world. When you see bubbles floating around you, your normal reaction is to pop the bubbles with your hands. When you see yourself

---

on top of a snowboard that starts sliding downhill, your reaction is to take control of the snowboard and try to avoid obstacles by leaning left and right. When you see yourself inside a pinball game, the thing to do is to keep hitting the ball into the game when it comes within reach. Just to be sure though, at the beginning of each game, a two-sentence voice announcement tells the player what to do.

Inter-game navigation uses the same physical world metaphors as the games themselves. For example, exiting a specific game is done by standing in front of a clearly marked exit sign that is out of the way of regular play.

Having a virtual world based on real-world situations creates instant familiarity. Since human body motion is subject to the laws of physics, so should the behavior within the virtual worlds, at least to some degree. Again, the physical behavior creates familiarity and meets intuitions of how things work. Hence, we have the tenth principle, P10.

P10: Virtual Worlds have Familiar Physical Behaviors

It would, however, not fully exploit the potential of a PC-based gaming system if the virtual environment merely simulated the physical world to the letter–especially in a game. Certain actions can trigger behaviors that are simply impossible in the real world but contribute tremendously to the magic of the experience. Being shrunk and captured inside a bubble, in the *Bubble Mania* game, and seeing yourself float away is magic; moreover, it exploits the full potential of the designer's imagination and the medium. Furthermore, it adds an element of discovery for the player who is eager to find out what will happen if he does either this or that. This brings us to the eleventh principle, P11.

P11: Virtual Worlds have Surprising Behaviors

The amount of floor and airspace around the home computer can cause confusion for the computer vision technology. The games were designed for a single player in the camera's view at any one time. Moreover, the vision algorithm is optimized for a single head and, at most, two hands. If more than one player appears in the camera's field of vision, the computer will randomly alternate amongst the visible body parts. This can cause the game to malfunction in various ways, such as game objects not correctly interacting with the player or the player's position randomly jumping around the screen. We, therefore, have the twelfth principle, P12.

P12: Only One Player at a Time

While some of these principles may seem obvious, enthusiastic application designers may get carried away. They might assume that computer vision interaction technology can deliver far more powerful user experiences than it can in practice. The Me2Cam game designers at the Smart Toy Lab know; they've been there. During our exploration, numerous activity ideas had to be abandoned because they ultimately violated one or more of these principles, which at the time, we had yet to discover and articulate.

CONCLUSIONS

"*It's the application, stupid*" was a popular catch phrase around Intel's technology labs for a while. It was used to remind technologists and engineers that technology itself is not important; what ultimately matters is what the technology is used for. This paper provides an application analysis case study, from the application designer's, not the implementer's viewpoint for the Intel® Play™ Me2Cam* Virtual Game System.

It is the application designer's role to understand the strengths and weaknesses of the underlying technologies and to take an unwavering human-centered perspective in defining the experience. In the mind of a skilled application designer, a well-designed application cannot promise and deliver more than what the technology can provide.

Computer vision will never replace today's user interface, definitely not for current applications. However, when used appropriately, it does have a place. It can either augment existing interfaces or, as this paper illustrates, provide different experiences ("new uses, new users") altogether.

ACKNOWLEDGMENTS

The authors thank the toy designers Brett Bogar (Mattel) and Frank Brown (Mattel, now Intel), Anne Paper (Mattel), Melanie Goldstein (Mattel, now Intel), Kendal Miller, Terry Ycasas, Bill DeLeeuw, Judy Goldstein, Cory Cox, Raj Hazra, David Linn, the people from the now-defunct ePlanet: Emily Albertson, Jamie Christini, Katie Franz, and John Gomes, and many others for turning the idea of a computer vision game for children into a shipping product.

REFERENCES

www.intelplay.com

---

http://www.intel.com/ial/home/digentertain/immerse.htm

AUTHORS' BIOGRAPHIES

Herman D'Hooge manages the Design Center for Intel's Connected Products Division. Prior to that, he managed New Concept Development at the Intel Smart Toy Lab which he co-founded in early 1998. During his 20 years with Intel he has held various positions in research, development, product architecture, industry evangelism, and management in areas ranging from multi-processor computer architectures, PC system architecture, operating systems, computer security, fault tolerance, distributed systems, computer telephony integration, and new computing applications research. He holds an M.S. degree in Electrical Engineering and an M.S. degree in Computer Science, both from the University of Ghent, Belgium. His e-mail is herman.d.dhooge@intel.com.

Michael Goldsmith is a Staff Software Engineer in Intel Labs. He currently works on turning Video-As-Input technology into real-world applications. During his 10 years at Intel he has held various positions in development, software architecture, and management in areas ranging from operating systems, compilers, speech-based applications, PC-based video conferencing, and Video-As-Input applications. He is the creator of the Pinball activity. He holds a B.S. degree in Computer Science from the Massachusetts Institute of Technology. His e-mail is michael.a.goldsmith@intel.com.

Audio Software for the Intel® Play™ Computer Sound Morpher

Scott Boss, Smart Toy Lab (Connected Products Division), Intel Corp.

Index words: Consumer Products, Extended PC, Smart Toys, Signal Processing, Audio

ABSTRACT

The Intel® Play™ Computer Sound Morpher (CSM) is a hand-held audio recording device paired with easy-to-use, PC-based, sound-editing software. It is targeted at children aged eight to twelve and allows them to explore the world of sound in fun and creative ways. As is true of all Intel® Play™ toys, the CSM consists of a bundled hardware device and software suite with its own unique set of implementation challenges. This paper outlines those challenges and presents the implementation details and approaches involved in building this audio-based Smart Toy.

From early concept prototypes to final product, the CSM software evolved from a collection of toy designer dreams, user interface sketches, and audio technologies into a unified multimedia application for children. Several of the original prototypes and experiments are presented to set the stage for the final product. Both the hardware and software evolved through stages of prototyping in order to find the best mix between technical capabilities and design requirements.

This paper focuses on the audio software technologies needed to provide a full software feature set and to make those features usable for children. This behind-the-scenes look at the CSM software reveals approaches to and implementation details of several features including audio tone detection, energy detection, singing text to speech voices, visual representations of audio, and audio effects filters. The technical details of the audio components developed at the Smart Toy Lab (STL) are presented along with the integration effort needed to tie them and other third-party solutions into a software application. This paper also presents the architectural decisions made in order to balance the tug of war between required software features and the time and resources available to make them part of the product.

In the end, the audio software for CSM was feature packed, consisting of several third-party vendor components as well as several Smart Toy Lab-developed audio components. Together, these software pieces combined to present a novel approach to exploring and manipulating sound using a personal computer. When combined with the CSM hardware, the software becomes part of a complete PC play experience that brings new uses and a new perspective to the PC.

INTRODUCTION

As the hardware for the Computer Sound Morpher (CSM) took shape and became simpler in its function, the need to add significant play value with the PC became evident. The onus fell on the software to deliver a broad, open-ended play experience to help deliver the Intel Play brand promise. In this paper, we describe the solutions that were put in place to deliver a large feature set on a tight schedule.

The CSM software consists of a collection of audio technologies bound together to present a rich feature set within a child's multimedia application. The audio technologies were developed both in-house at the Smart Toy Lab and out-of-house by several external vendors. Some technologies were built from the ground up and others were licensed and integrated as-is from third parties. The combination of existing components and newly developed components presented a significant integration and validation challenge.

CHALLENGES

The challenges presented by the Computer Sound Morpher (CSM) software stemmed mainly from the breadth of the application. Individually, the sound technologies were well known; however, collectively they presented a large integration challenge. In particular, the main challenges were as follows:

- the coordination of several software development teams
- the integration of external software components
- delivering a robust feature set implementation

In total, there were eight separate software groups both within Intel and external to Intel contributing components to the CSM software effort. Three of these groups were actively developing new code as the project progressed. Mapping out and coordinating the software development of these groups involved substantial planning and monitoring. Deliverables were tracked, integrated, and evaluated continuously as the project progressed.

The end product required a seamless presentation of the various software features. The application developer was presented with the large task of integrating these components. To aid in this effort, the Smart Toy Lab (STL) invested time and resources in developing a library to combine the third-party components and present them to the application developer as a single unit. This library not only simplified the integration of components into the application, but it also served as a vital tap into the application software stack for debugging and problem-solving purposes.

Delivering a robust feature set was a broad effort including the use and development of several in-house and out-of-house components. Each component was individually validated and tested before being integrated into the application. Some of the third-party components were currently shipping products that were simply integrated as-is and only required some code to drive them from the application. Making sure these components were feature-complete and functioned without error was an involved task that was met by the STL integration team.

FROM SNOOPERS TO PHASERS

Early development of the Computer Sound Morpher (CSM) resulted in many hardware and software prototypes, each playing a critical part in the evolution of this toy. By taking a look at this history, the reader can better understand the objectives of this project and the tradeoffs made in order to produce the end product.

While there were several hardware prototypes developed, the common theme throughout the early exploration stage was capturing "fun" sounds. The first attempts focused on capturing sounds from a distance in order to deliver an exciting play pattern, i.e., listening to sounds not easily heard with the naked ear. The prototype delivered on this promise with a slick "deflector dish" design that collected sounds from a range of 50 to 80 feet in an extremely directional manner. The sound quality and sound-collection ability of the prototype was tested by wiring the microphone and dish assembly to an evaluation sound-recording board. Sounds were recorded, transferred to the PC, and then analyzed for sound quality.

Figure 1: First functional prototype

The first prototype (Figure 1) provided solid data for determining desired sound quality and sound-collection ability. Unfortunately, its form factor was much too large. The next prototype faced the challenge of meeting two significant constraints.

- a tight material cost budget
- a small form factor

Figure 2a: Collapsible dish prototype–closed

The collapsible dish prototype (Figure 2a and 2b) incorporated the sound-capturing capability of the first prototype, but added a folding petal feature to reduce the form factor. Sounds could still be collected from a long distance when the petals were extended and the toy could easily fit in a small package when the petals were collapsed. Its shortcoming, however, was cost of manufacturing and complexity of design. The folding petals were not easily constructed and required complexities in manufacturing that were error prone and expensive.

Figure 2b: Collapsible dish prototype–open

Thus far, each successive prototype was a refinement on the previous one: each one maintained the ability to capture sounds at a distance and at the same time improved on the form factor by virtue of it being more compact. In fact, the prototypes were victims of their own success when long-distance sound capture became a stumbling block due to its snooping and eavesdropping connotations. The final prototypes and eventual end product took on a sleek, high-tech look that emphasized the compact lines and minimized manufacturing complexities and cost.

The final engineering prototype (Figure 3) was constructed to validate the audio recording components in terms of form fit and functionality. This final prototype was fully functional as a recording device and used final production parts. As such, this prototype helped the toy designer in constructing a final form factor for the toy and helped the team determine the final cost of materials for the toy.

Figure 3: Engineering final prototype

Figure 4: Intel® Play™ Computer Sound Morpher

After the completion of the hardware prototyping, the final result was the CSM product that appears on store shelves today (Figure 4). It is the culmination of all the learning discovered during the prototyping and design stage. Ultimately, the expensive, long-distance sound capture device was traded for a low-cost, compact listening device that is now the Computer Sound Morpher.

The software exploration and prototyping was conducted in parallel with the hardware prototype development. The purpose behind the software prototypes was to explore, evaluate, and gauge the complexity of proposed software features. For the CSM, these features included the live voice changer, sound filters, synthetic voices, and audio streaming infrastructure. The CSM prototype integrated these features into a simple demonstration that was used for the following purposes:

- to allow software engineers to evaluate third-party software components
- to foster a dialog between engineering and design on the feasibility of features
- to help define sound filters by being used as a tool by designers
- to validate STL-developed components later in the project.

AUDIO CORNUCOPIA

The software feature set of this toy is packed with many fun activities centered on capturing, creating, and manipulating audio. These activities allow the child to explore the world of sound and include the following:

- **Sound download–**Detect and segment sound recordings from the hardware and store on hard drive.
- **Visual displays–**Present static waveform plots of captured audio for editing, and display live visual animations, as audio is played back.
- **Animated talking head–**Take any audio from the software library and play back the audio synchronized with an animated talking head.
- **Audio cut and paste–**Remove words or audio chunks from any recording for placement into a second recording.
- **Live voice changer**–Modify live audio input, i.e., the child's voice, from the toy hardware with sound filters and route the modified audio back out the speakers.
- **Sound filters**–Transform and manipulate any recorded or live audio stream. This included standard filters such as echo, reverb, and chorus as well as specialized filters such as 3D sound and noise reduction.
- **Synthetic voices**–Generate audio in one of several voices from text typed in by the user.
- **E-mail**–E-mail a sound recording by itself or connect it to an animated face.
- **Sound clips**–This is a built-in library of canned sound clips included on the software CD.

Behind each one of these features is some sort of audio technology to bring the feature to life. These technologies range from signal-processing algorithms and text-to-speech engines to waveform plotters and visual-effect generating libraries. At the core of all these audio technologies is a streaming infrastructure to tie everything together and route audio where it needs to go.

SOFTWARE ARCHITECTURE

The various audio technologies were bound together and presented as a seamless multimedia application. This section describes each of the pieces of this software puzzle and how all the pieces of the puzzle fit together. With all the pieces in place, it becomes clear how each piece met a specific need of the toy feature set (Figure 5).

The Computer Sound Morpher (CSM) software architecture followed the typical Smart Toy Lab (STL) product approach. The software was divided into three main categories: the application, the middleware, and the drivers. Fortunately, for the CSM, off-the-shelf audio drivers were used since the device connected to the PC via a standard audio input jack on the existing PC sound card. This left the middleware and application layers to be resolved by the STL.

The application layer, including the user interface and framework, was outsourced to a third-party multimedia application development house. This development house was experienced in developing rich multimedia applications for children, but lacked some of the specific technology needed for the CSM feature set. To fill in the missing pieces, other third-party packages were licensed including:

- text-to-speech engine
- audio streaming library
- audio sound effects library
- visual display library

- noise reduction library
- 3D sound algorithm

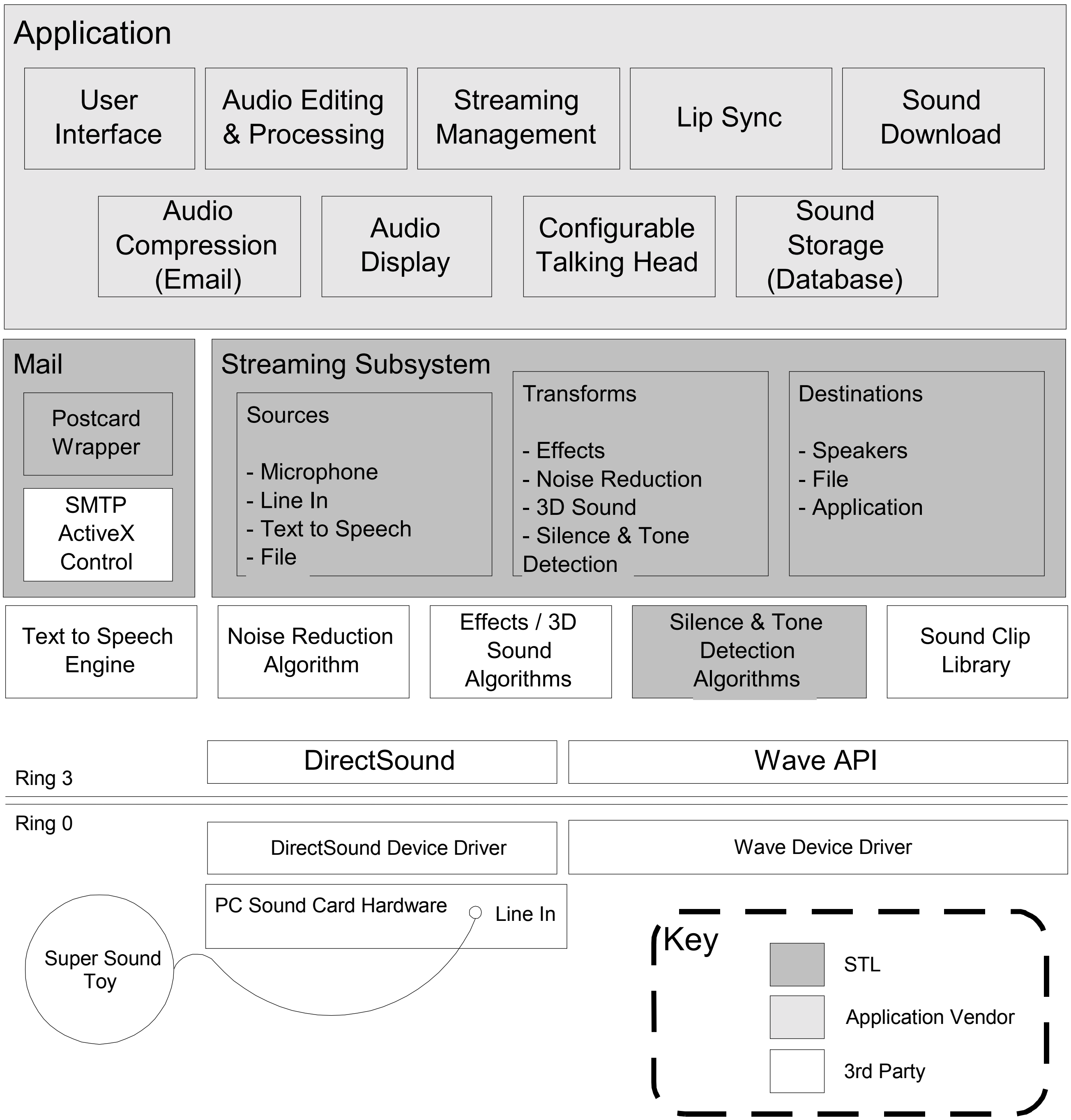

Figure 5: Computer Sound Morpher software architecture

Because of the variety and number of additional third-party pieces involved, a middleware component was developed at the Smart Toy Lab to unify all third-party pieces and present them as a single unit to the application developer. This "glue" provided a single point of access and simplified interface to the application vendor for all the technologies involved.

In addition to tying together third-party components, the Smart Toy Lab also provided some signal processing algorithms to implement involved features such as segmenting audio, recognizing hardware inserted tones, detecting audio feedback, and changing audio pitch.

Finally, there were other software pieces borrowed and modified from another software peer group within the Consumer Products Division. The Create & Share Camera team provided technologies to configure audio settings for playback and recording volumes as well as libraries to automatically connect to the Internet and send mail.

EYE CANDY

Putting a user interface on an audio application is an interesting task. This was accomplished by creating visual representations of sounds, which were both static and dynamic in nature. Whenever an individual sound was downloaded, edited, or created, a waveform plot was displayed (Figure 6). This waveform plot provided a visual representation for the audio that could be manipulated and examined.

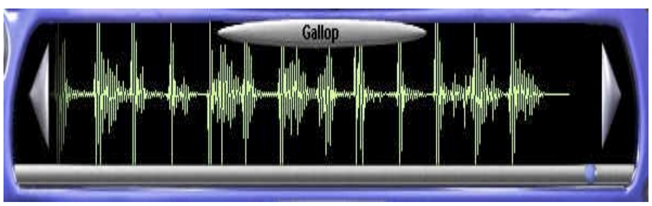

Figure 6: Static waveform plot

In addition to the static waveform, another third-party solution was licensed to provide dynamic visualizations of the audio (Figure 7). As audio was played back from file, or routed by the live voice changer, or created by the text-to-speech engine, it was also displayed as an animated graphic. The third-party software included several different visualizations, all of which provided an active on-screen experience while audio was being manipulated.

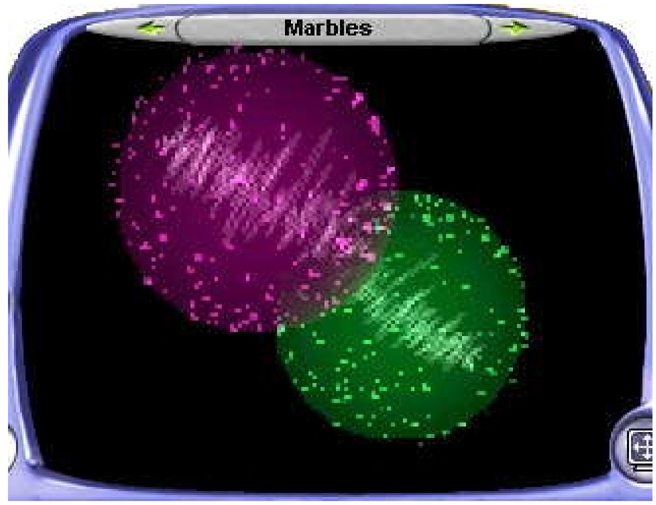

Figure 7: Visual display

CREATING ALIENS AND MONSTERS

With its main purpose as a sound-creating and sound-editing tool, the software naturally required several technologies to analyze and manipulate audio. Features such as downloading sounds from the toy, cutting up sounds into pieces, streaming live audio from the toy to the speakers, and applying special effects to sounds all required direct analysis and manipulation of the audio. Specifically, the following technologies were used in order to implement the feature set listed previously:

- **Frequency Shift Keying data modulation**–Used to download and delineate multiple captured sounds from the hardware.
- **Edge detection**–Used to identify and extract portions of audio from captured sounds.
- **Digital signal processing**–Used for special effects such as echo, reverb, chorus, and pitch as well as 3D sound and noise reduction.
- **Feedback detection**–Used to detect feedback during live voice loop back from the toy to the speakers.

Bring It Down

Downloading sounds from the hardware was constrained by the one-way analog connection from the toy to the PC via an existing sound card. There is no mechanism to "communicate" with the hardware from the PC due to this analog connection. Starting and stopping sound download, as well as segmenting individually captured sounds, required a unique communication mechanism. Existing and well-known modem technology was applied to overcome this limitation.

The hardware used Frequency Shift Keying data (FSK) modulation in order to encode information about the captured sounds into the analog data stream flowing from the toy to the PC. The encoded information describes the data on the toy in enough detail for the software to successfully start and stop the download of sounds to the PC as well as to check for aborted downloads by the toy. The information includes the number of sounds and the total length of all sounds on the hardware as well as sound delimiters to mark the beginnings and ends of sounds. This information was stored on the hardware and was readily accessible by the firmware. The user initiates the transfer of sounds from the toy to the PC by entering the sound download screen in the host software and then pressing the download button on the toy (Figure 8).

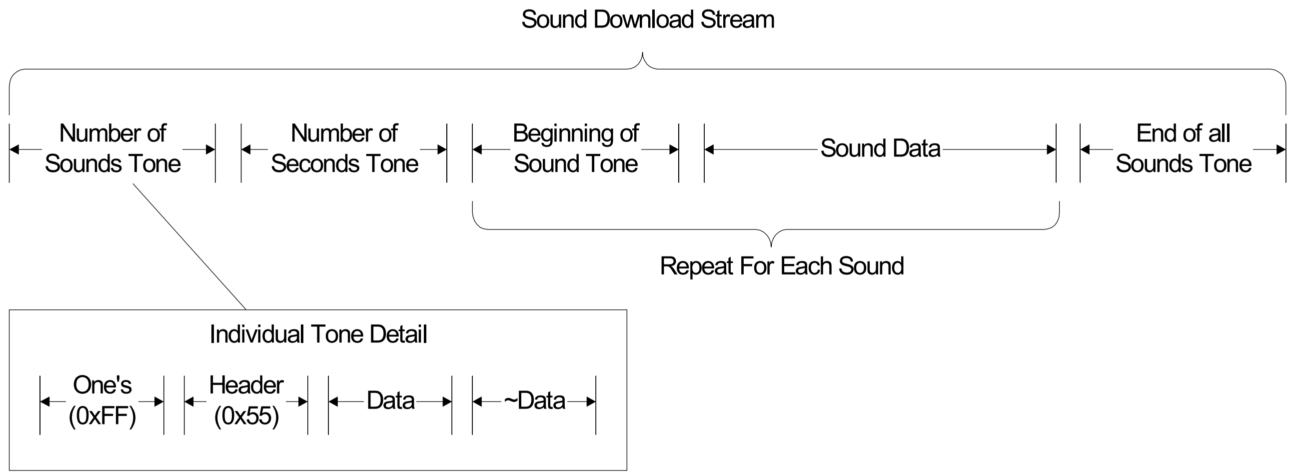

Figure 8: FSK download stream format

By entering the sound download screen, the user indicates to the software that a download will possibly follow. The software then opens an active input channel to the hardware via the PC sound card and monitors this channel for the FSK tones from the hardware. After the download button on the toy is pressed, the hardware encodes the number of sounds and the total length of all sounds in seconds as FSK tones in the analog output going to the computer.

The number of sounds and total length of all sounds in seconds designate the beginning of a sound download from the hardware. The software uses these two pieces of information to determine whether the following sound data constitutes a valid download. Following the initial information, each individual sound is preceded by a beginning-of-sound tone, and the last sound is followed by an end-of-all-sounds tone. Individual sound lengths are not encoded in the stream; however, the length of an individual sound is determined by the beginning-of-sounds and end-of-all-sounds tone markers.

The software detects and decodes each of the tones in order to maintain a download state machine. Each of the tones mentioned above must be present in order to complete a download. In addition to detecting tones, the software also overwrites the tones with silence before sending the audio to the speakers. Each individual sound is extracted from the live audio stream and written to a unique file.

The individual tones themselves each consist of a sequence of four characters or bytes of information. The length of the characters is determined by the baud rate of the modem, which is approximately 302 bits per second (bps). Each character consists of 10 bits (first bit = 1, 8 bits of data, last bit = 0) and, with the given baud rate, results in a character length of 33 milliseconds. To ensure that random audio data does not get interpreted as a valid tone, the tone format was defined as follows:

- Character 1 consists of a sequence of 10 bits all set to one.
- Character 2 consists of the tone header with data part equal to 0x55 (0x1010101010).
- Character 3 consists of the data for the tone.
- Character 4 consists of the opposite of the data in Character 3 (each bit is flipped).

Encoding each tone as four characters with a header, data, and opposite data values makes it extremely unlikely to detect random tones inside the valid audio data.

Cut It Up

The "cut and paste" feature of the software required analysis of sounds for "boundaries" or changes in energy level. Once identified, these boundaries defined portions or chunks of audio that could then be extracted from one piece of audio and placed into another piece of audio. An edge-detection algorithm was developed at the Smart Toy Lab to implement this feature. Targeted mainly at detecting words within spoken sentences, this algorithm also served the purpose of breaking up generic sounds into segments (Figure 9).

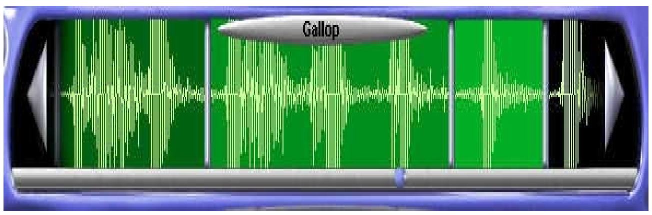

Figure 9: Segmented audio display

The algorithm was designed to work in two stages. The first stage walks the entire buffer of audio samples and calculates energy levels to be used during the second stage. The second stage then uses the pre-calculated energy levels to identify and return sound "boundaries" to the application. A sound "boundary" is a drop in sound energy with specific constraints and can be controlled by the application through a minimum feature size parameter. This parameter allows the application to generate several "cut up" versions of the audio and present them to the user on demand. This proved to be very useful for breaking up speech into phrases, words, and even syllables.

Morph It

Audio manipulation was the name of the game for the CSM software. To bring this "morphing" to life, a wide array of special filters was provided for sound manipulation. These filters were used in several of the features, including the live voice changer, the synthetic voices, and the sound-editing area. To do this, the filters were integrated into an audio-streaming system that provided for application of filters to live audio from the sound card, to audio output from the text-to-speech engine, and to audio from a file.

The majority of the special filters was provided by a third-party software vendor and included standard audio algorithms such as echo, reverb, chorus, phaser, and flange. The algorithms were combined and configured in order to create unique effects, which were presented to the user as a list of voices. For example, a voice simulating a group of aliens consists of a combination of chorus and echo algorithms to generate multiple space-like voices. The third-party software provided the infrastructure for combining the algorithms, which proved to be a powerful and flexible framework for creating new effects on the fly.

To add to the fun, a special third-party algorithm was licensed to provide a 3D audio effect. This algorithm was packaged and presented to the user in the identical way the core algorithms listed above were presented; however, this algorithm was unique in its complexity. It exposed parameters to allow for positioning of the audio in space, which the application could manipulate over time to simulate the sound moving from left to right or from in front of the user to behind the user.

In addition to the algorithms provided by the third-party vendors, a couple more were added at the Smart Toy Lab to allow for more audio manipulation. These included pitch and volume algorithms, which allowed for the creation of chipmunk- and monster-like voices as well as for the adjustment of the volume level of individual recordings.

Finally, the ability to clean up audio recordings was provided via a noise reduction algorithm developed in the Intel® Architecture Lab. This noise reduction was again presented as another option in the list of effects or voices and allowed the user to remove unwanted hum or hiss from captured recordings. Because of the portable nature of the CSM hardware, just about any type of recording could be made in any type of recording environment. With this in mind, the noise-reduction algorithm gave users a tool to remove an unwanted audio from their captured recordings.

Feed It Back

The live voice changer feature of the software allowed children to speak into the CSM hardware and then hear their voices come back out the speakers in some modified form. This live loop back of audio presented a unique set of challenges including the following:

- how to minimize audible delay from the microphone to the speakers
- how to maintain an uninterrupted audio stream
- how to detect and handle audio feedback

The audible delay and uninterrupted audio stream issues were at constant odds with each other. In order to minimize delay, small audio buffers are required so that the first chunk of audio can be collected from the sound card and immediately sent to the speakers. Unfortunately, small audio buffers are much more likely to result in interruptions in the audio streaming due to system activity. If the audio thread does not receive processing time for a period greater than the audio buffer size, then the audio stream is interrupted and a pop or click results. The utilization of Direct Sound and its primary buffer support allowed for low latency in the audio stream and minimized the delay. However, much tuning and compatibility testing was required in order to find a consistent sound setting across all audio hardware.

For anyone familiar with audio feedback, it is evident that placing a live microphone in front of a pair of speakers is not always a good idea. High-pitched squealing and wailing is often the result. For the live voice changer feature, this was a problem. A dual approach was taken in order to minimize this problem since it was out of the scope of the project to completely eliminate it with a technology such as echo cancellation.

First, the user was guided through an audio set-up wizard upon installation of the software in order to select the "ideal" sound card input and output volume levels for the live voice changer. Once configured, these settings were used by the application and significantly reduced the chances of producing audio feedback while using the live voice changer. Second, an algorithm was written at the Smart Toy Lab to detect when audio feedback occurs. Once the feedback was detected, the application was notified and could make the necessary adjustments to help the user fix the problem. The application turned the volume down to zero and then instructed the user to slowly turn up the volume to an acceptable level without producing feedback.

The feedback-detection algorithm was very simplistic in its approach but proved to be effective. It measured feedback as a high-energy period in the live audio stream. Energy was measured in decibels, and if a certain decibel threshold was maintained for a minimum time period, then this was determined to be a feedback situation. It is possible to generate false feedback detections in this case, but this was deemed more desirable than allowing feedback to occur unchecked.

Talking PC

Creating a recording from typed-in text was yet another fun feature of the software. The user was presented with a list of voices to choose from which included all kinds of wacky characters. The text-to-speech engine provided the guts behind this feature and was extended and combined with special effects to create wacky custom voices. The text-to-speech engine was licensed from a third party and was selected for its ability to configure and create new text-to-speech voices as well as for its capability to provide raw audio buffers to the client.

The creation of text-to-speech voices involved a few different technical approaches. First, new voices were created simply by adjusting the built-in parameters of the text-to-speech engine itself. These parameters were adjustable using in-line text command sequences, which were interjected before the text typed in by the user. The types of voices generated using this approach include child voices, female, male, and high- and low-pitched voices.

Second, the output from the text-to-speech engine was sent through the various special effects algorithms to produce an even wider range of voices. This is where the voices started to take on the wacky form where space creatures, cockroaches, and monsters came to life.

Finally, a couple of advanced features of the text-to-speech engine were utilized to create "singing" voices. These voices allowed the user to type in text and hear the words output as a melody. For example, imagine the phrase "Singing words is big fun," sung back to the tune of "Happy Birthday To You."

The approach taken to allow for a generic phrase utilized the phoneme and pitch features of the text-to-speech engine. The engine was configured to generate a stream of phonemes from the provided text. Phonemes are parts of speech and can be thought of as portions of a syllable. This stream of phonemes was then broken down into sonorants and non-sonorants. The sonorants are the audible portions of words such as vowel sounds. Each sonorant was assigned a pitch and duration in order to generate the singing sequence. The entire phoneme stream was reconstructed as a new text sequence with the pitch and duration commands embedded in the sequence. This encoded phoneme stream of text was then fed back into the text-to-speech engine, which generated the actual "singing" audio.

TYING IT ALL TOGETHER

With so many audio components involved, an efficient and simple interface was needed to present the various technologies to the application developer. A single component was developed at the Smart Toy Lab to house the various audio technologies and present a single interface to the application. This audio component also managed the routing of audio to and from the various pieces, which included the necessary multi-threading to allow for prioritization of audio over the visual displays and user interface code of the application.

The audio library implements the streaming of audio from several sources through a set of transforms, and then to one or more destinations. A third-party audio streaming infrastructure was used as a base to provide low-level direct sound, wave, and audio mixing support. This helped to isolate any driver specific support in a single place. On top of this streaming infrastructure, the audio library encapsulated the various technologies as follows:

- **Audio sources**–These are file sources, text-to-speech sources, live microphone sources, and memory sources.
- **Audio transforms**–These are basic effects algorithms (echo, chorus, reverb, etc.), Smart Toy Lab algorithms (FSK tone detection, edge detection, volume, pitch, feedback detection), special algorithms (3D sound, noise reduction).
- **Audio destinations**–These are file destinations, speaker destinations (via audio mixer), application destinations.

Audio can be routed from any source, through any transform, and to any destination either individually or simultaneously. Each routing is referred to as an audio stream, and the only limitation on the number of audio streams is CPU load and memory. The following list contains the example usages of the sound library by the application for various features.

- **Sound download–**The audio is streamed from the microphone source, through the FSK tone-detection algorithm, and routed to the speakers for playback, to the application for live video display, and to individual file destinations for storing the downloaded files.
- **Sound editing–**The sounds captured by the user are loaded into memory for editing and previewing. The user can also request that they be run through the edge-detection algorithm or audio special effects transforms. Finally, they are routed to the speakers for playback, to the application for visual display, and, optionally, back to file if users request a save for their edits.
- **Live voice changer–**The audio is streamed from the microphone source and routed through special effects transforms and the feedback-detection algorithm and then to the speakers for playback and to the application for visual display. Users also have the option of sending the audio to file if they want a live voice recording.
- **Synthetic voice generation–**The audio is generated by the text-to-speech source and routed through special effects transforms and then to the speakers for playback and to the application for visual display. Again the audio is also routed back to file if users request a save for their edits.
- **User interface sounds–**The user interface sounds are first read from file and stored entirely in memory, as they are small and need to be played back immediately. The sounds are then mixed with any other active audio and then sent to the speakers.

Creating the sound library involved a lot of plumbing, and required constant integration into the application along with constant debug and refinement. Many delivery dates and checkpoints were set up at the beginning of the project to support application development while this audio library came to life.

EXTENSIBILITY

Much of the software was designed to allow for content updates after the product shipped, without modifications to the application itself. This allows for new application content to be delivered, such as add-on packs or updates, after the toy is out on the market. The areas that are extensible include the following:

- audio special effects
- text-to-speech voices
- audio visualizations
- noise-reduction configurations

Each of these areas can be configured by adding or modifying entries in the Windows* registry.

SUMMARY

The Computer Sound Morpher (CSM) software brought together a large set of audio technologies into a single fun-packed child's application. A lot of effort from several sources went into the implementation of the feature set, which resulted in a rich and diverse final product. This paper focused on the audio-processing portion of the CSM, which was a large chunk of the application. However, the complete application involved more than just processing audio. Development of the user interface with various screens and features

---

*Other brands and names may be claimed as the property of others.

was very involved. Each of the features described in this paper were bolted up and exposed to the user in one form or another. In the end, the final software product was a successful integration of sound technologies, user interface components, and colorful artwork and animations.

ACKNOWLEDGMENTS

Mark Leavy was instrumental in designing and implementing several of the signal-processing portions of the Computer Sound Morpher. Specifically, the approaches behind the sound-download, edge- detection, and feedback-detection features were outlined and brought to life by Mark.

AUTHOR'S BIOGRAPHY

Scott Boss is a software engineer who has been with Intel for eight years and has been working at the Smart Toy Lab for the past two years. Before the Toy Lab, Scott worked in the Intel Architecture Labs on various multimedia projects ranging from audio infrastructures to 3D graphics. He received his B.S. degree in Computer Science from Valparaiso University in 1991 and his M.S. degree in Computer Science from Purdue University in 1993. His e-mail is scott.d.boss@intel.com.

Technology and Play Pattern:
Intel® Play™ Digital Movie Creator

Michael M. Conti, Smart Toy Lab (Consumer Products Division), Intel
Riley Jackson, Smart Toy Lab (Consumer Products Division), Intel
Henry Bruce, Smart Toy Lab (Consumer Products Division), Intel

Index words: Consumer Products, Extended PC, Smart Toys, Digital Video Processing, Audio

ABSTRACT

The flagship product for Intel Play for 2001 is the Digital Movie Creator (DMC), which is one of the many consumer products that came out of the Connected Products Division at Intel. The DMC is a hand-held digital video and still camera paired with easy-to-use, PC-based, movie-creation software (Figure 1). It is targeted at children, ages eight and up.

Figure 1: Digital Movie Creator

From early prototypes to final production, the DMC evolved through many stages of development: market research, usability studies, user interface design, and engineering feasibility studies. These stages helpto identify cost and implementation risks. We needed to ensure that we could deliver a $99USD product within budget and on schedule.

This paper provides a high-level outline of the technology drivers behind this unique movie-creation toy. These were the four drivers of the movie-making aspect of the product:

- image quality
- audio quality
- video frame rate
- audio video synchronization

We look specifically at how these technology drivers defined and were defined by the play pattern.

We also look at how end-user's expectations set some of the requirements for the technology drivers. And finally, we suggest what steps should be taken for future products that will ensure the proper compromises are made between technology and play patterns.

INTRODUCTION

The Intel® Play™ Digital Movie Creator (DMC) is a low-end digital alternative to using traditional film or camcorder video cameras. Instead of exposing film to light, the user is exposing light to an image sensor. Instead of playing back a film or videotape, the user is retrieving images from digital storage. It is a dual-mode digital camera that operates as both a digital camcorder and a digital still camera. It interfaces to a PC (Intel® Pentium® or Celeron® Processor MMX, 300MHz or better) via a USB connection. The camera may be operated in two modes: untethered from the PC (battery powered) or tethered, as a full-speed USB peripheral.

End-User Expectations

To determine an acceptable quality level is at best subjective. The determination of quality rests in large part on what the user's expectations are. Children can be a lot less demanding of quality than adults who are familiar with more high-end equipment. If the end-user is expecting quality standards commonly found in a Sony Hi8* camcorder when they use the DMC, they are bound to be disappointed. If the product delivers to their expectations for a children's toy, then the team has made the right compromises in choosing the technology drivers.

It was determined early on that the primary play pattern of the DMC is the capturing of content, either as movie clips or still images, and then playing them back on the PC. To deliver that experience, expectations for the product were that it *must* provide "good" quality in the following areas:

- image and audio
- video playback frame rate
- sychronized audio video

TECHNOLOGY DRIVERS

The five technology drivers to be discussed here are image quality, frame rate, audio quality, video-audio synchronization, and storage. Many of the technology drivers are dependent upon what choices the team made in their selection of hardware components.

Image Quality

Image quality can be affected by many variables, including lens selection, frame rate, and insufficient lighting. Poor design or a lack of attention to any one of these variables can result in poor images.

The toy needs an image sensor to capture snapshots and video clips. There are two flavors of image sensor: CCD and CMOS devices. CCD sensors have better performance; this is particularly noticeable in home lighting. The picture is sharper and less grainy and the colors stronger. However, CCD sensors are more expensive than CMOS image sensors. Due to the cost constraints, a CMOS sensor was used for the DMC and therefore image quality was sacrificed.

F-Stop and Lens Selection

Another variable in the image quality equation is the lens. A lens assembly is used to focus light onto the image sensor. The assembly is made up of a number of lens elements, which are made of either plastic or glass. Glass lens elements perform better, mainly because they let through more light and give a crisper image. However, plastic lens elements are considerably less expensive. Lens assemblies can be comprised of glass and/or plastic elements. A combination of element types is called a hybrid lens.

Also, the lens focus is fixed, as there is no way of checking focus when the DMC is untethered. Our decision to fix the lens focus was validated by the observation that children are not familiar with manual focus. Most children are familiar with simple "point-and-shoot" cameras.

Once again, a compromise was made due to cost, and the team decided to use a plastic lens. An f2.8 lens was used to ensure that the camera has a large depth of field. This was deemed to be more important than the greater light sensitivity offered by a lower f number lens. The f numbers indicate the size of the aperture (opening) relative to the focal length (distance from camera to subject). f numbers are calculated by dividing the focal length of the lens by the effective diameter of the aperture, e.g., 55mm lens, effective aperture 5mm = relative f11. Focus group feedback showed us it was better to make the camera work well in a variety of light settings with some loss of image quality.

This choice of the f stop is one of the compromises that came out of the play pattern usage model and which supported the secondary play pattern, stop-motion animation.

Stop-motion animation requires the user's explicit need to be close to the target that is being animated. Stop- motion animation involves the end-user manipulating an object in close range of the camera and taking frame-by- frame snap shots. The single frames are then combined into a single video (.avi) file.

Frame Rate

Frame rate isn't normally an issue when shooting in the traditional formats as video is fixed at 30 fps (frames per second) and film is 24 fps.

We looked at various frame rates with the DMC; both untethered and tethered operations, trying to decide from a user's standpoint what would be the best frame rate. It is important to note that in the digital format, a higher frame rate requires a faster processor and more storage space. Higher video frame rates would result in less available total record time (untethered). The team chose a lower end frame rate of 10 fps to provide the most flexibility to the product's play pattern.

Video Processing and Compression

Converting the raw data that comes from the image sensor into images requires complex Digital Signal- Processing (DSP) algorithms. Typically there are two forms of ASIC architecture that achieve this task: a programmable DSP core or hard-coded logic. The former is more flexible but more expensive, and it is used in products such as the Kodak mc3*. For the DMC, a "hard-coded" ASIC was used. An example of the compromise made here was that the ASIC had a bug that reduced the dynamic range of the processed image. If this had been a programmable device, the issue could have been corrected, but as the functionality was hard coded, this option was not possible.

To maximize the untethered recording time, the video has to be compressed. There are a number of algorithms suitable for this, and selecting one is usually a tradeoff between complexity, compressed image quality, and compression ratio. For the DMC, the Joint Photographic Experts Group (JPEG) compression algorithm was used, as this algorithm can be implemented as logic, resulting in lower ASIC cost. While the JPEG is more typically used for still image compression, for video each frame is JPEG compressed. This is termed Motion JPEG or MJPEG. Moving Pictures Experts Group (MPEG) is another video compression technique. It also compresses the difference between two subsequent frames (inter-frame compression), which results in a higher compression ratio at the cost of increased implementation complexity and cost.

Resolution

Although the image sensor has 352x288 pixels (known as CIF or Common Interchange Format), we used the central 320x240 Region Of Interest (ROI), as it allowed us to record 30% longer recording time without severely impacting overall image quality.

Audio Quality

Audio capture plays an important role in the creation of any movie. The DMC's play pattern supports this activity; it has a microphone built into the housing. While people can tolerate low-quality video, poor audio is much more noticeable and distracting. Thus, a significant effort was made to keep audio quality high. Microphone placement, digitization, and audio compression were areas that affected audio quality.

Microphone

An omni-directional microphone was used. Although a directional microphone is more desirable, it would have been too expensive. The microphone was positioned in the housing to ensure a maximum range of six feet, with minimal pick-up of handling noise and seismic rumble.

*Other names and brands may be claimed as the property of others.

Digitization

An AC97 compliant codec was used to digitize the signal from the microphone. Although this was "overkill" from a technical perspective, the ASIC has an AC97 input port, so this was a simple solution to implement.

The audio gain was set at the device's highest gain setting and stored in the internal SRAM. While it would have been better to provide an automatic gain control, this wasn't possible with the current design. Instead, we realized the user's desire to capture audio from at least four feet away and supported that with the high setting. When watching a movie, you notice the sound of someone's voice doesn't necessarily rise and fall based upon his or her distance from the camera, as this would be distracting. However, for the DMC, we had to compromise: the closer you are to the DMC, the louder you will sound and the further away you are, the softer you will sound.

Audio Compression

As for video, to maximize the video clip record time, the audio data should be compressed. Adaptive Differential Pulse Code Modulation (APDCM) compression was used for the normal setting. For the high-quality setting, the microcontroller executing the firmware did not have enough power to compress the audio, so the data were left in Pulse Code Modulation (PCM) format.

Video Audio Synchronization

As we know, a movie that is out-of-sync is pretty distracting. Some foreign movies, which are over-dubbed in English, come to mind: the action of the lips doesn't follow the words being heard.

In filmmaking, sound is usually recorded separately with specific sync points being generated (the reason for the "clap" board) and the end-result is synchronized in post-production. This final playback to film is generally done with optical audio tracks or magnetic tracks that run along the filmstrip. For the DMC, the end result needed to be the same as in film, i.e., the picture and audio had to be in sync, but the process of getting those two elements captured and played back wasn't the same.

Given that the application needed homogenous audio of 22KHz 16-bit mono PCM for simplified editing, the driver components responsible for download of audio from the camera to the PC up-sampled the audio stream. As mentioned above, the audio stored on the camera was in two different formats depending on the quality setting used when the audio/video was captured and stored. If the camera was in normal quality/resolution mode, then the audio was stored uncompressed at a 12KHz sample rate. If the normal quality/resolution setting was used, the audio was stored with a custom form of DPCM compression and sampled at 8KHz. Thus the host software took this into account when doing the sample rate conversion to the desired audio format prior to the creation of the final .avi file

Another benefit of allowing a quality choice was in the extension of the movie clip time: lower quality equals longer record times.

Integrating post capture sound in a movie clip happens a couple of different ways. In one scenario, the end-user captures someone singing Happy Birthday and then hears and sees it at her computer later. In another scenario, she could decide that the person singing Happy Birthday was doing a terrible job, so she mutes that voice and re-records over the same scene with her own. In yet another scenario, she could decide that the singing is fine but that it needs some background music. These opportunities to manipulate the post capture sound expanded the play pattern.

Storage

The camera requires some form of storage for snapshots and video clips captured when untethered. The cost of storage would limit us to two technologies: Synchronous Dynamic Random Access Memory (SDRAM) or FLASH (a marketing term of fast programmable EEPROM, or Electrically Erasable Programmable Read Only Memory). At the time of printing, SDRAM was one-fifth of the price of Intel StrataFlash®, so it was the chosen technology.

SDRAM is the most common type of computer memory; also known as D-RAM or DRAM. It usually uses one transistor and a capacitor to represent a bit. The capacitors must be energized hundreds of times per second in order to maintain the charges.

Unlike Flash, SDRAM needs an amount of current in order to retain its memory when "off," so a compromise was made to provide an off-on switch through button presses. In other words, the user had to first press a button to wake up the camera after it had gone to sleep. Then he had to press the button again to initiate the desired action, taking a snap shot or shooting video.

Video frames were captured to a temporary SDRAM buffer in CIF or QCIF resolution in YUV420 format. A more detailed discussion on YUV is presented in the hardware architecture section below.

Snapshots are video clips stored in SDRAM by using a simple filing system. A 64-byte header block precedes each item. For a snapshot, this is followed by the JPEG data, which is written to SDRAM directly by the JPEG compressor. For a video clip, the header is followed by interleaved audio and video data. The avi data comprises video frames with associated audio samples. Each frame is preceded by a 16-byte frame header, and is followed by the JPEG frame and then the audio samples that were captured during that frame time (copied from the internal SRAM). Due to restrictions of the filing system, no single data "block" can be larger than 32kbytes. This only affects 320x240 pixel JPEG snapshots, which could be of higher quality without this restriction. The frame header effectively acts as a time stamp so that audio and video data are synchronized.

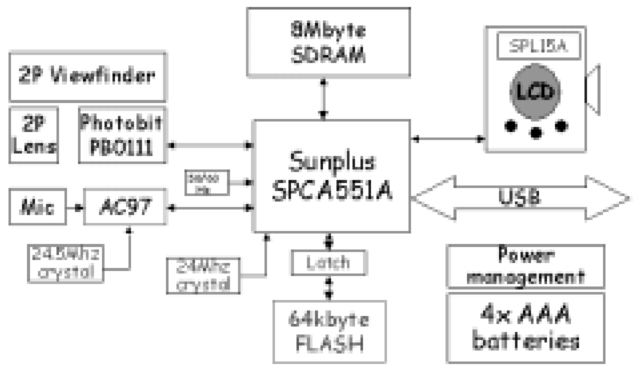

Figure 2: Hardware architecture

HARDWARE ARCHITECTURE

By discussing the technology drivers for component selection, we have almost covered the hardware architecture itself.

The largest, and most expensive component is the Application Specific Integrated Circuit (ASIC). Think of the ASIC as a chip that is custom designed for a specific application rather than a general-purpose chip such as a microprocessor. ASICs are "hardwired" to do a specific job and do not incur the overhead of fetching and interpreting stored instructions as in a computer. An ASIC chip performs an electronic operation as fast as it is possible to do so, providing, of course, that the circuit design is efficiently architected.

Some ASICs do the processing and compression of YUV data to JPEG data in firmware, but they need expensive processors or they are too slow. Doing the work in hardware makes the ASIC cheap, but you have to "approve" the processor blocks, as you can't change them.

The ASIC also interfaces to the rest of the components (e.g., image sensor, audio codec, SDRAM, etc.). The Sunplus SPCA551A was chosen as it met most of the play pattern requirements for a very competitive price. The

---

Intel StrataFlash is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

ASIC's central role in the hardware architecture is shown in Figure 2. An embedded 8052 microcontroller core executes the firmware out of external FLASH program's memory. This microcontroller is a minor Achilles heel for the ASIC: it was not powerful enough to handle the video data and compress the audio in the high-resolution mode.

The ASIC picked by the team utilized a Complementary Metal Oxide Semiconductor (CMOS) image sensor. Note that most video camcorders use a Charge Coupled Device (CCD) rather than CMOS. The ASIC colorizes the images, compresses the data using JPEG, and then stores them to memory or uploads them to the PC. To maximize video clip time (untethered) SIF (320x240 pixels) and QCIF (160x120 pixels) image sizes are used.

Although the CMOS sensor produces 352x288 (CIF), we used 320x240 (SIF) as it allowed us to record 30% more data into the camera's 8 megs of Synchronous Dynamic Random Access Memory (SDRAM).

Firmware

The complexity of the firmware was driven by the camera operations in the untethered mode. For instance, the whole system goes to sleep when not in use. It was a major firmware challenge to wake up, adjust exposure and white balance, and take a snapshot all in under two seconds (see Figure 2).

In the tethered mode, the firmware was primarily concerned about keeping the video stream going and tracking button presses.

SOFTWARE ARCHITECTURE

Figure 3 describes the breakdown of various software components that make up the DMC application and software stack.

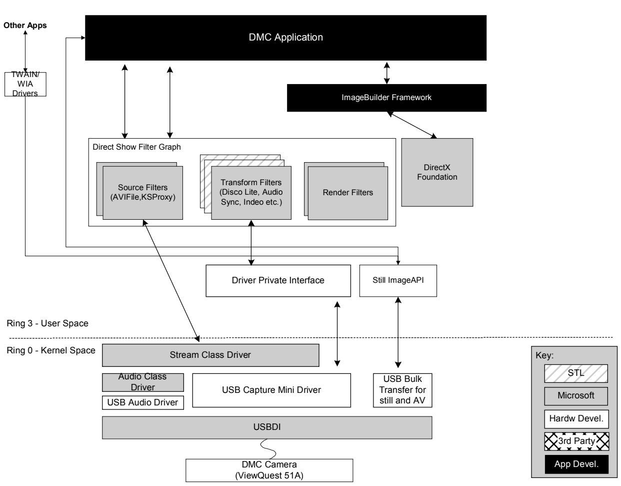

Figure 3: Software architecture overview

The application was responsible for creating and configuring the various DirectShow filter graphs for real-time capture and processing of audio/video data. Custom interfaces were used to interface with driver extensions implemented for specific DMC hardware that went beyond the scope of a typical camera device. A custom playback engine was developed for layer movie playback.

DirectShow*

The workhorse of the live video system was the Microsoft DirectShow* filter graph module. The filter graph was responsible for obtaining the audio/video stream from the DMC device and enhancing/converting each frame and then rendering it. This was accomplished by passing the video stream through several downstream filters in the filter graph. DirectShow provided an extremely flexible architecture in which filters are "plugged" into the graph to provide unique and specific behaviors.

The application interacted and controlled the DMC USB device via DirectShow's Capture Filter interfaces. A custom interface was exposed to the application, if needed, to allow it access and control over extended camera features.

Figure 4 below, describes the main filter graph that was configured by the application for preview and capture of video and snapshots. The camera provided I420 format video at CIF resolution (320x240). This video stream was then directed downstream into the AVI Decompressor, which produced RGB24. The Image Interpolation (Scale Image) filter was used to scale to video stream to a larger resolution for live preview at a larger size (1.3xSIF).

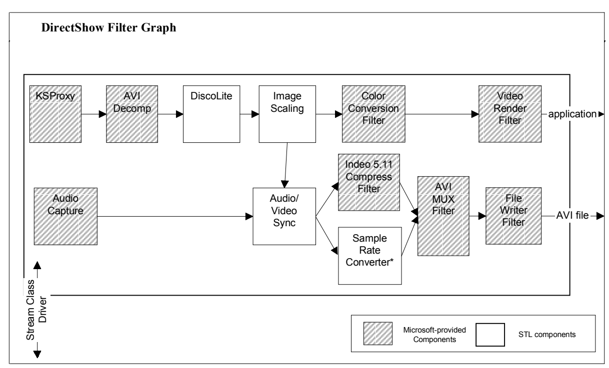

Figure 4: DMC "Live" Filter Graph

The camera also provides MS-class compliant audio. The in-camera USB microphone appears to DirectShow and

* Other brands and names may be claimed as the property of others.

Indeo is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

the application as a standard audio capture device. The audio stream sample rate, bit depth, and channels are converted to the desired format by the Audio Sample Rate Converter filter.

Compression

To conserve hard disk space, all AVI files were compressed using Indeo® 5.11. Video compression was done in the capture stream of the graph via the Indeo 5 real-time compressor.

Direct Show Filter Detail

Below are descriptions of the individual filters that comprise the various filter graphs.

KSProxy* Source Filter

The KSProxy source filter was provided by Microsoft and provided a generic interface to streaming video capture devices. It supported interfaces allowing access to standard camera properties and control (brightness, contrast, saturation, hue and video format, etc.).

Audio/Video Sync Filter & Mux

The audio/video filter used in conjunction with the Microsoft DirectX* 8 (DX8) MUX filter provide a reasonably reliable, single synchronized stream from the audio and video input sources. The audio/video sync filter's main task was to ensure the streams' start and stop points where synchronized. The MUX automatically drops and inserts video frames based upon the audio master stream and specified frame rate.

Image Interpolation Filter

The image interpolation filter was used to scale the video frames as they are sent downstream. This is a standard transform filter. This filter uses simple bilinear interpolation. Use of this filter provides a more dependable mechanism for image scaling when compared to the Microsoft video renderer filter, which can be video card dependent.

DiscoLite* Transform Filter

The DiscoLite transform filter first acts as a pin splitter. It takes the capture output pin from the KSProxy filter and splits it, thus allowing the application access to the two video output pins: capture and preview. These pins are then being controlled in a "gated" manner. If the gate is raised then the video stream flows downstream. If the gate is closed, video is not sent downstream. This provides the application considerable flexibility when controlling the filter graph when it is implementing features such as video capture. All pins on the DiscoLite filter (input, preview, capture) handled only RGB24 as a video subtype.

RESULTS

DMC's hardware and software architecture as described show that it is possible to engineer a product within the constraints of its technology drivers, and still provide a movie-creation experience. However, did DMC deliver to the user's expectations? Probably not if that person was expecting something similar to a home video camera. And to be honest, the quality just isn't there in terms of image and audio quality. But as a toy, it does deliver. The digital format holds much promise at the higher end products, but to justify a $99 USD price point, compromises were made in the selection and implementation of the drivers. These compromises were as follows:

- The image quality was lowered by the choice of a 2.8 f, which provides a wider variety of lighting situations during play.
- Audio-video synchronization is not ideal, but is acceptable to the end-user without adding to overhead performance on application side.
- Audio quality is acceptable for normal play pattern distance, 6 feet or less, but lacks auto gain controls and frequency controls for low-end sounds.
- There is a low frame rate, but it provides maximum ability for storage on the device.

At the end of the day, image and audio quality become less important. Children enjoyed the accessibility of the USB-enabled product and also seeing the immediate results of their work. Other movie-creation features not discussed here, such as sprite animations and painting on the video, also added value.

DISCUSSION

It is important to note that technology drivers enable the play pattern but don't make the toy. That experience comes from the end-user's experience with everything else, from the form factor to the other movie-creation activities. All of these items have equal importance in the final product. While Intel doesn't lack in its engineering ability, it must not lose sight of the big picture: to deliver to, and exceed, the consumer's expectations.

Here are some things that worked when designing the Digital Movie Creator (DMC).

- Evaluate the consumer's expectations with the product that is being built.

*Other brands and names may be claimed as the property of others.

For instance, we used focus group testing to determine if four minutes of untethered capture time was enough. A previous year's product was only going to use eight seconds!

- Make sure that the product has features that go beyond the primary feature–capturing movie clips.

  For the DMC, a strong secondary feature was the ability to do stop-motion animation and the fact that it had an array of creation tools.

- Observe the end-users interaction with the product.

  Do usability studies, and listen to the children. When image quality was an issue, it was generally thought to be the fault of the computer not being fast enough rather than the toy itself.

CONCLUSION

Technology is only one of the ingredients that make a consumer product great. How that technology is applied to an everyday task or activity and its relationship to a desired play pattern, makes a toy entertaining, fun, and unique. The Digital Movie Creator is a good example of a toy that takes the technologies of video capture and editing and applies them to storytelling, thereby creating a whole new category of play.

ACKNOWLEDGMENTS

Thanks to Jim Okuley (toy designer) for being the play pattern guru and everyone else on the DMC team who helped make this product a reality.

AUTHORS' BIOGRAPHIES

Michael M. Conti has been with Intel at the Smart Toy Lab since 1999, and is currently a program manager with CPD. Michael has 15 years of production management experience, from movie making to computer games. Michael has a B.A. degree in English. His e-mail address is Michael.M.Conti@intel.com

Riley Jackson is a Sr. Software Engineer and has been with Intel since 1991. He has been a software architect/developer at the Smart Toy Lab for two years. Past assignments include Intel Architecture Labs, Pro Share, and the Development Tools Org. He has 20 years experience in application and component design and development. His e-mail is riley.jackson@intel.com

Henry Bruce is a hardware engineer who has been with Intel at the Smart Toy Lab for a year. Before Intel, Henry worked for VLSI Vision Ltd. (VVL) in Edinburgh, latterly acquired by STMicroelectronics. There, as software development manager, he developed expertise in video processing and device drivers. He received his B.S. degree in Electronic Engineering in 1985 and his Ph.D. degree in Image Processing in 1993, both from the University of Edinburgh in Scotland. His e-mail address is henry.bruce@intel.com.