

Microsoft Robotics Studio in Education

Stewart Tansley, PhD

Program Manager

Microsoft Research

<http://research.microsoft.com/~stansley>

Contents

- What is MSRS?
- What's available today?
- What's different about MSRS?
- Who's MSRS for?
- MSRS for Education
- A word on MSRS in IPRE

Microsoft Robotics Studio

- A development platform for the robotics community, supporting a wide variety of users, hardware, and application scenarios

Microsoft® **ROBOTICS STUDIO**

Runtime

- Concurrency
- Services infrastructure

Authoring Tools

- Simulation Tool
- Visual Programming Language

Services and Samples

- Samples and tutorials
- Robot services
- Robot models
- Technology services

<http://www.microsoft.com/robotics>

MSRS today

- v1.0 released December 2006
- v1.5 imminent (July 2007)
- ~65MB download (recent beta 1.5), ~10 min install
- No cost for academic and non-commercial use, full product, not an “eval version”!
- No MSDNAA membership required – though that’s a no-brainer to join! ☺
- Runs on Windows & .NET
 - *But your robot certainly does not have to!*
- Tested and supported to work on C++, C#, VB, Javascript, IronPython
 - *But any .NET language should work, including F#, Scheme, Smalltalk, etc.*
- >100 services already built-in, including generics, simulations, robot platforms:
 - iRobot Create, Pioneer, BoeBot, fischertechnik, KHR-1, Lego NXT & RCX, ... (no constraints on others)
- Many 3rd party services emerging for other robots and functions:
 - Scribber, SRV-1, VEX, Traxster, Lynxmotion, ER1, Phidgets, KUKA ... (see web for more)
- Comprehensive walkthrough-style tutorials (>30), videos (8)
- Active community site
- >100k downloads to date
- ~40 commercial partners
- At least 10 active academic research projects known (but we don’t have a way to track this today)

What's different about MSRS?

- **Core runtime:**
 - Managed code (.NET, Java, etc.) abstracts memory management and certain security concerns from the programmer
 - MSRS adds *concurrency* and *distribution* to this list
 - Provides a pervasive services-oriented paradigm for development
 - REST architecture – the abstract paradigm of the web
http://en.wikipedia.org/wiki/Representational_State_Transfer
 - A way forward for the programmers of tomorrow?
 - Leverages your existing web tools and skills (browser, XML, etc.)
- **Optional** modern game-quality 3D simulator, with physics, complete transparency between sim/real implementations
- **Optional** Visual Programming Language entry point to development, complete transparency between VPL/native code
- **Industrial-strength, commercial-ready platform, free for educational use**

REST

(Representational State Transfer)

- “RESTafarians” argue that the Web scaled so well as a direct result of key design principles:
 - Application state and functionality are divided into **resources**
 - Every resource is uniquely addressable using a **universal syntax** for use in **hypermedia links**
 - All resources share a **uniform interface** for the transfer of state between client and resource, consisting of
 - A constrained set of **well-defined operations**
 - A constrained set of **content types**, optionally supporting **code-on-demand**
 - A protocol that is:
 - **Client/Server**
 - **Stateless**
 - **Cacheable**
 - **Layered**
- REST's client-server separation of concerns simplifies component implementation, reduces the complexity of connector semantics, improves the effectiveness of performance tuning, and increases the scalability of pure server components.
- Layered system constraints allow intermediaries for further scalability — proxies, gateways, and firewalls.
- REST enables intermediate processing by constraining messages to be self-descriptive:
 - Interaction is stateless between requests, standard methods and media types are used to indicate semantics and exchange information, and responses explicitly indicate cacheability.
- **Roy Fielding** defined REST in his doctoral thesis in 2000 – and is one of the principle authors of HTTP
- As is **Henrik Frystyk Nielsen** – Tim Berners-Lee's first graduate student, and a key architect of MSRS

Who's MSRS for?

(in its native form, straight out of the box)

- **Commercial**
 - Robot application developers
 - Robot platform vendors
 - Robot component suppliers
 - Potential: non-traditional robot applications
- **Academic**
 - Educators who want explore teaching programming with robotics, perhaps especially REST-style programming
 - Researchers who want a “plumbing layer” for robotics
 - Researchers who want a simulator for robotics
 - Researchers who want to extend MSRS for additional scenarios, e.g. traditional beginner programming, non-robot devices
- **Hobbyists**
 - Sophisticated homebrew systems with an emphasis on software
 - Tinkerers, Explorers, Kids, Parents, Everyone!
- Note: trivial use of MSRS in its native form is **easy**. But to use its full power, sophisticated software development skills, especially with .NET (or Java) experience, is necessary and there is a learning curve. This is why so much effort has been put into tutorials and support materials – and more will follow, e.g. books, third party education materials. Feedback?

MSRS for Education (excl. research; primary scenarios only)

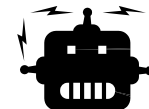
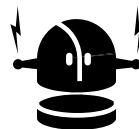
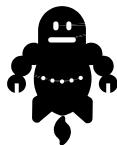
(Disclaimer: Stewart's opinions)

- **Teaching programming/CS at university level and advanced HS**
 - MSRS ready to use in certain classes today
 - 30 tutorials to leverage – easy port to curriculum contexts, e.g. labs
 - But it's a sophisticated professional software platform, so best for advanced level courses (cf. full Visual Studio & .NET)
 - Good where you want to teach the REST paradigm, differentiated from conventional web programming
 - Flexible hardware options, with ready-to-run services for common educational robots already “in the box”
 - Neat 3D simulation option to engage students exposed to computer games
 - Visual Programming Language is an easier entry path for less experienced programmers
 - MSRS was not designed for novice traditional programmers per se, but we're interested in feedback from instructors
- **Teaching beginner programming at university level and HS**
 - Same as above
 - However, the complete infrastructure may be overwhelming at this level
 - One raison d'être for IPRE (Georgia Tech/ Bryn Mawr) – we are researching this very problem space!
 - IPRE expects to deliver a thoroughly-proven teaching-specific software solution, wrapping MSRS complexities and adding pedagogical features demanded by the beginner programmer class
 - Unless you want to explore “REST-first”, perhaps – feedback please
- **Teaching engineering/robotics**
 - Same as above
 - Again, the full software infrastructure may be daunting for engineers with shallow software skills
 - Basic use of the platform capabilities very possible though, using higher level tools such as VPL and the simulator
 - Can provide a thorough and flexible plumbing layer for advanced robots
 - Possibly a way to introduce REST-style programming to engineers – feedback please



A word on IPRE & MSRS

- **IPRE**
 - The Institute for Personal Robots in Education, <http://roboteducation.org>
 - Hosted at Georgia Tech with Bryn Mawr College
- **Vision**
 - A personal robot for every student programmer
 - A broad community of practitioners in educational robotics
- **Goal**
 - Rigorously research the effectiveness of a special purpose personal robot in the CS1/CS2 classroom, with the aim of increasing attraction and retention
 - Share the results of assessment and materials so that others in the community can adopt the approach
- **MSRS role in IPRE**
 - A software tool to provide hardware abstraction, programming language independence, additional capabilities, & accelerate the development time
 - Myro is the name of the software platform for IPRE
 - Metaphor: Myro has “MSRS inside”
 - Put another way: Myro is a “pedagogical wrapper” for MSRS



Microsoft[®]

Your potential. Our passion.[™]



© 2007 Microsoft Corporation. All rights reserved.

Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.