SANTOS

Open Architecture
API and Plug-ins
Unique Underlying Simulation Structure
- Joint-based model
- Optimization-based model

Open Software Architecture

Software Expansion

Generalized Interface for External Applications
- Text Files
- Dynamic Data Exchange (DDE)
- Networking

API
- Documentation
- Manuals

SDK
- Examples
- Templates & Wizards

Plug-in Development
- Directly with C++ or C#
- Using scripting language
  (Python, LUA, etc.)

Add to Software

Link with External Capabilities

Software Integration
Component overview
Software Integration
Current Santos Architecture

Santos(tm) Architecture

Plugins
- Task Plugin
  - C# version 3.0
  - Dynamics EXE
    - Native code executable
- Visual Plugin
  - C# version 3.0
  - Profile Graphs
    - Display and analysis
- System Plugin
  - C# version 3.0
  - Biomechanics
    - Biomechanical human model
  - Cognition
    - Add cognitive capabilities
  - Physiology
    - Advanced physiology model

Documented API

MDI Application EXE
- C# version 3.0
  - Forms/Controls
    - Controls, forms, DockPanel DLL
  - Built-in Forms
    - List window, tool panel, library, etc.
  - Sequencer GUI
    - Interface for basic sequencer rendering
  - Property Controls
    - Property control factory system

G.O.D. System DLL
- Executes subtasks in the app
  - Manager
    - Loads and manages plugins
  - Undo System
    - Stores G.O.D. actions to undo/redo

C# Core DLL (CsCore.dll)
- C# DLL interacts with the Core DLL for the MDI app, plugins and other DLL's
  - Core Library
    - Vector, matrix, color and math support
  - Messaging
    - Shared memory messaging system
  - Asset Property
    - Asset Property System
  - ID System
    - ID referencing system for objects
  - World System
    - Packaged objects, and the scene graph
  - Object Properties
    - Carbon object property factory
  - Project System
    - Loads and saves projects, asset and sequences
  - Sequencer
    - Sequences tasks and manages animation

VsCore DLL (VsCore.dll)
- Standard C++ DLL interacts with Virtools and unmanaged C++ libraries
  - Core
    - Global support functions
  - Shared Memory
    - Global memory, flags & variables
  - Posture IO
    - Interface with posture prediction library
  - Virttools Support
    - Manipulate & set the Virtools world

Posture Lib
- Standard C++ posture library
- Posture
  - Uses optimization to predict posture

Virttools Lib
- Standard C++, Virttools Script & managed C
  - Virttools Sim
    - VSL and Building Block Script
  - Virttools Player
    - Managed C++
• **Plug-in Architecture**
  • Offers robust & powerful API
  • C# DLLs using wizards and full SDK
  • Most challenging for developer to implement

• **External Applications**
  • Use intermediate input/output format
  • Modules can be written in any language
  • More restrictive & less efficient

• **Scripting Language**
  • Allows quick development in familiar language
  • Can be done directly from SANTOS™ (no compiling)
  • Limited API but customizable
Static Joint ROM Plugin

Plugin interfaces to Santos through our plugin system.

The plugin then uses our Core API to access 80% of all program functionality.
Software Integration
Plug-in Architecture

Analysis Toolkits
- Spine Shear & Compression
- Physiology Tools
- Strength Analysis
- Etc...

Custom Objects
- Armor
- Stairs
- Ladder
- Etc...

Tasks
- Posture
- Dynamic Walk
- Stair Climb
- Etc...

Systems
- Editing Schemes
- Custom GUIs
- Clothing Suite
- Motion Connect
- Etc...

SANTOS™

Done!
• Communicate through intermediate formats
  - Text files
  - Dynamic Data Exchange (DDE)
  - Networking

• Can accommodate more rigid or legacy modules

• Currently used for Predictive Dynamics, Zone-Diff, etc.
Allows quick creation of custom modules using a common language such as Python, LUA, etc.

Requires:
- Integration of a scripting language and interpreter
- Implementation of a robust API
Thank You