COLLADA: An Open Standard for Robot File Formats

Rosen Diankov (University of Tokyo), Ryohei Ueda (University of Tokyo), Kei Okada (University of Tokyo), Hajime Saito (General Robotix)

1. Why COLLADA?

The robotics industry is overflowing with open and proprietary robot file formats. It seems that every major program and library has its own format and own set of quirks. Furthermore, no matter how hard a developer attempts to unify the robotics information, there will **always** be information that another developer wants to insert. Although a lot of recent formats allow adding new information to extend the original content, there are very few formats that can manage new content and allow extensibility of existing content with the scalability required for major projects. Fortunately, there exists an XML-based open standard called COLLADA, the COLLAborative Design Activity [1] that is *content-scalable*.

The major characteristics that make COLLADA unique from other formats are:

- the assumption that the information format is continuously evolving and spans multiple files over multiple databases, and
- multiple organizations will want to specify the same information using their own standards.

It is the power of digital content management that allows COLLADA to outshine any other ISO format [2]. COLLADA originally started as a 3D model format and is currently supported by many open source packages like Blender and OpenSceneGraph. The most recent version of the standard is 1.5 and includes both physics and kinematics extensions, which paves the way for robot-specific extensions. The AutomationML group [3] has already started creating an ecosystem of programs around COLLADA. And recently, many robotics libraries like ROS [4], OpenRAVE [5], OpenHRP3 [6], OpenGRASP [7], and EusLisp have developed a common set of robotics extensions and can import and export robots in COLLADA.

In this paper we describe a large amount of work that has gone into making COLLADA usable for robot descriptions. We review the COLLADA features, the current work towards developing robot extensions for COLLADA, the current robot software that can understand COLLADA, and a robot database where the most popular robots can be downloaded.

2. COLLADA Features

The COLLADA specification is divided into:

- core content management tags dictating the referencing structure,

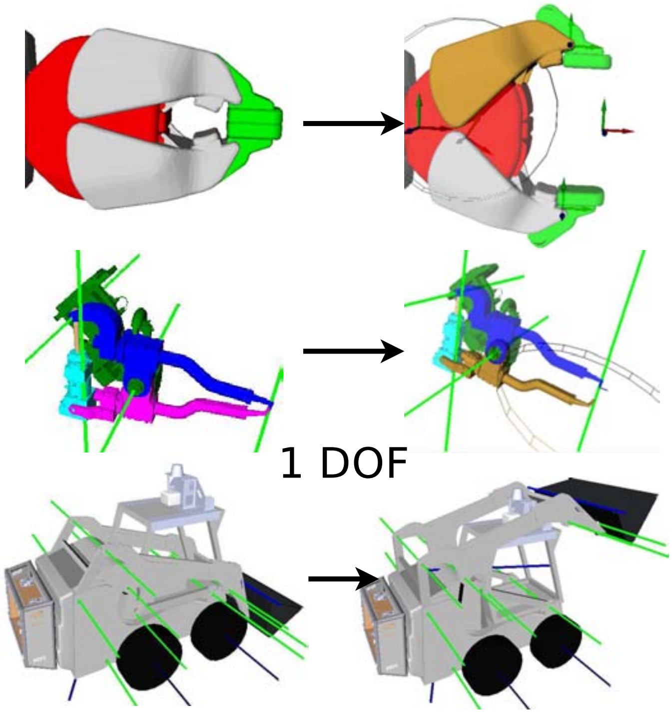

Fig.1 COLLADA allows complex kinematics with closed loops to be easily written.

- graphics scene extensions allowing complex shaders and geometries,
- simple rigid-body dynamics parameters,
- boundrary representation models,
- and kinematics structures.

COLLADA's referencing system allows libraries of geometries, effects, kinematics bodies, physcis bodies, and joints to be saved in a different file and referenced in a separate COLLADA file through URIs. This added flexibility makes the COLLADA Specification seem daunting and very difficult to work with at first, but by viewing it as a mini-language embedded inside the XML schema, users should be able to quickly grasp the core concepts.

COLLADA defines different types of scenes for graphics, physics, and kinematics which can function independently from one other. Users **bind** joints and links of the individual parts of each seen together in order to form one coherent environment. A very difficult concept to grasp is that there sometimes is not one-to-one relationships between graphics and kinematics. The graphics scene graph is mostly controlled by coordinate frames and effects while the kinematics scene is controlled by links and joints. Sometimes a complex avatar can have a very simple underlying skeleton represented by kinematics. One of the most powerful features of COLLADA kinematics is that robots with complex closed links can be specified (Fig. 1) using the MathML standard.

3. Robot-specific Extensions

In order to full specify a robot, we take full advantage of the kinematics specifications introduced in version 1.5. From our experience with OpenRAVE development, it was clear what parts critical to robot automation were still lacking, therefore we introduced a new set of robot extension tags [8]:

- **manipulator** - holds a subset of the joints in a robot and defining a hand and an arm,
- **sensor** - holds camera, laser, encoder definitinions and where to place the sensor on the robot,
- **actuator** - hold the actuation model of each joint,
- **collision group** - allow a robot to have multiple collision models separate from visual information.
- **mimic equations** - hold MathML equations for the partial velocity and accelerations of a mimic joint along with the position.

The extensions will continue to evolve and eventually our hope is that they will contribute to a new addition to the COLLADA standard. It is surprisingly easy to specify the extensions in a form that other users are familiar with and also provide explanations for the design choices made.

4. COLLADA Robot Database

Through OpenRAVE, we provide many sample robot models to allow users quick access to common robots and to show them how formats are built [9]. One of the biggest hurdles toward adopting a new format is the availability of compelling sample models for that format. The COLLADA robots can be readily used for motion planning and other simulations with no extra meta-tags.

Although COLLADA is XML-based, it has a compressed version of the format using the **zae** extension. This allows models to be compressed to 5% - 10% of their original size without impacting load performance as much.

OpenRAVE frequently synchronizes with the database and computes models like inverse kinematics solvers and measures their performance. The statistics for each robot are then uploaded on the web for everyone to share (Figure 2).

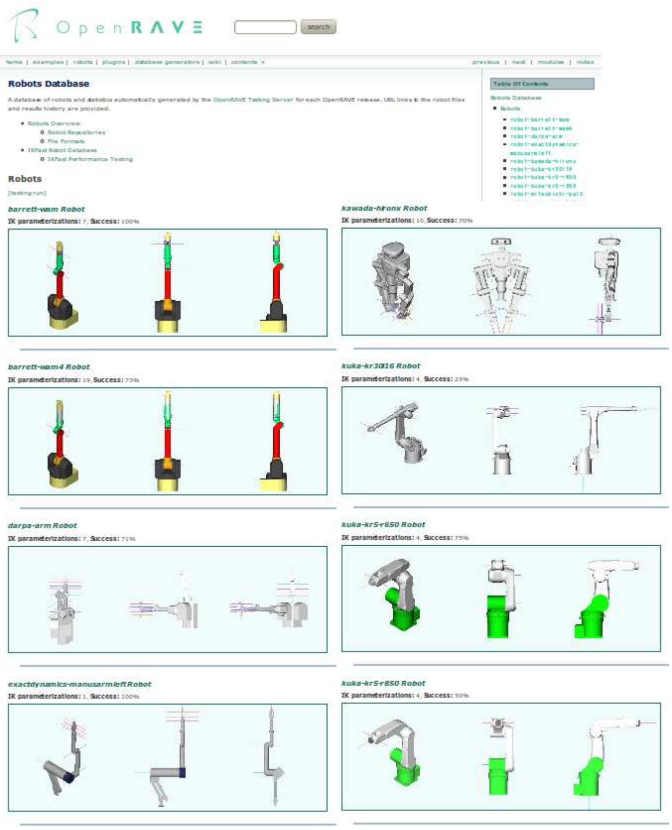

Fig.2 OpenRAVE provides a database of many robot models using the **OpenRAVE Profile** robot extensions.

5. Binding with Existing File Formats

Each major robotics tool has its own file format and makes it simpler for developers to work within in, but supporting tool-specific formats is really difficult since usually their are not standardized. Therefore, in order for COLLADA to gain a strong foothold in the robotics community, we manage importing and exporting COLLADA in some of the most common robotics tools in the research community. Each tool will continue to use its custom format for testing to new extensions, but at least data will be shareable. Figure 3 shows some of the major tools where importers and exporters are already in place and working at least at a basic level to provide model sharing.

5·1 OpenRAVE XML

OpenRAVE (Open Robotics Automation Virtual Environment) provides an environment for testing, developing, and deploying motion planning algorithms in real-world robotics applications. The main focus is on simulation and analysis of kinematic and geometric information related to motion planning. OpenRAVE's stand-alone nature allows is to be easily integrated into existing robotics systems.

The OpenRAVE XML file helps define manipulators and concatenate multiple robot parts into one more easily. Because the COLLADA robot extensions have been loosely based on the OpenRAVE API, most of the definitions in the XML file get exported into COLLADA. This makes it much simpler to create COLLADA files by exporting the OpenRAVE XML.

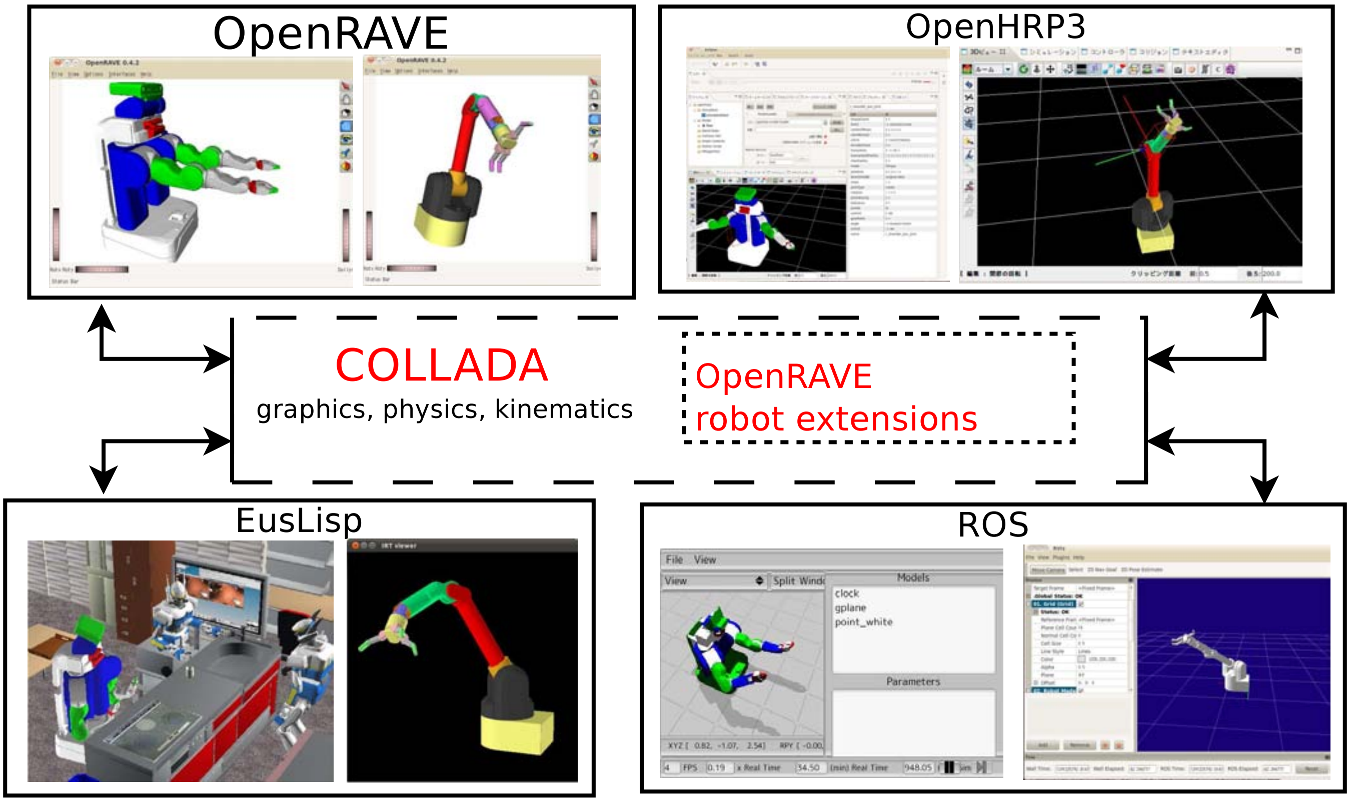

Fig.3 All package-specific file formats import and export COLLADA files. The converters for each software package are written.

5·2 OpenHRP3

OpenHRP3 (Open Architecture Human-centered Robotics Platform version 3) is an integrated software platform for robot simulations and software developments. It allows the users to inspect an original robot model and control program by dynamics simulation.

OpenHRP uses the VRML file format internally for managing robots, and the extensions are based on VRML PROTO structures. Once the VRML file is parsed, it is converted internally to an intermediate data format that can be sent through the network using CORBA. The COLLADA importer directly writes to the internmediate format and exports from the intermediate format.

5·3 ROS URDF

ROS (Robot Operating System) is vast collection of robotics packages that touch all parts of robotics. It is very common for individual researchers to create ROS wrappers for their libraries just so they can reuse ROS components and get their modules used by others.

The ROS team at Willow Garage created another XML-based format called Universal Robot Description Format (URDF). Fortunately, ROS is designed with very high modularity, so it was very easy to add COLLADA support through the **robot_model** ROS stack. The COLLADA exporter is an independent package within this stack called **colada_urdf**, while the COLLADA importer is directly embeded in the robot load methods and is invoked only if the file ends in **dae** or **zae** (compressed).

5·4 EusLisp

EusLisp is a Lisp dialict that has grown out of the JSK lab at University of Tokyo. The group recently open sourced the **jskeus** library that provides many humanoid motion planning functions and scripts. The EusLisp robot format itself is just a Lisp file that builds the robot as the file is executed. Although there are many advantages to using formats like this, it is very easy for users to extend the EusLisp robot format in a non-standard way and forget to tell everyone about the extensions.

Both importers and exporters are supported using the **euscollada** package.

6. Conclusion

COLLADA is an amazing file format with many content sharing capabilities that other formats could not even begin comparing with. The Khronos group managing the standardization of the format also provides conformance testing [10] for organizations who want to advertise COLLADA functionality. Regardless if COLLADA is currently not the mainstream CAD format like IGES and STEP, its flexible nature and powerful referencing can make it stand the test time. We will continue to push forward with the robot extensions into COLLADA and continue to provide as robot models as possible through OpenRAVE's Robot

Database.

References

[1] Khronos Group, "COLLADA Specification and Homepage." [Online]. Available: http://www.khronos.org/collada/

[2] R. Arnaud and M. C. Barnes, *COLLADA: Sailing the Gulf of 3D Digital Content Creation*. AK Peters, Ltd, 2006.

[3] AutomationML, "Standardizing Industrial Automation Formats." [Online]. Available: http://www.automationml.org/

[4] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "(ros): an open-source robot operating system," in *ICRA Workshop on Open Source Software in Robotics*, 2009.

[5] R. Diankov, "OpenRAVE: Open Robotics Automation Virtual Environment," 2006-present. [Online]. Available: http://openrave.org

[6] T. U. General Robotix, AIST Japan, "Open Humanoid Robot Platform 3." [Online]. Available: http://www.openrtp.jp/openhrp3/en/

[7] B. Le 坦 n, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moisio, J. Bohg, J. Kuffner, and R. Dillmann, "Opengrasp: A toolkit for robot grasping simulation," in *Simulation, Modeling, and Programming for Autonomous Robots*. Springer Berlin / Heidelberg, 2010, vol. 6472, pp. 109–120. [Online]. Available: http://www.springerlink.com/content/x043237l02605w50/

[8] Rosen Diankov and Ryohei Ueda, "Robot-specific COLLADA Extension Proposal." [Online]. Available: http://openrave.programmingvision.com/wiki/index.php/Format:XML

[9] OpenRAVE, "COLLADA Robots." [Online]. Available: https://openrave.svn.sourceforge.net/svnroot/openrave/data/robots

[10] Khronos Group, "COLLADA Conformance Testing." [Online]. Available: http://www.khronos.org/collada/adopters/