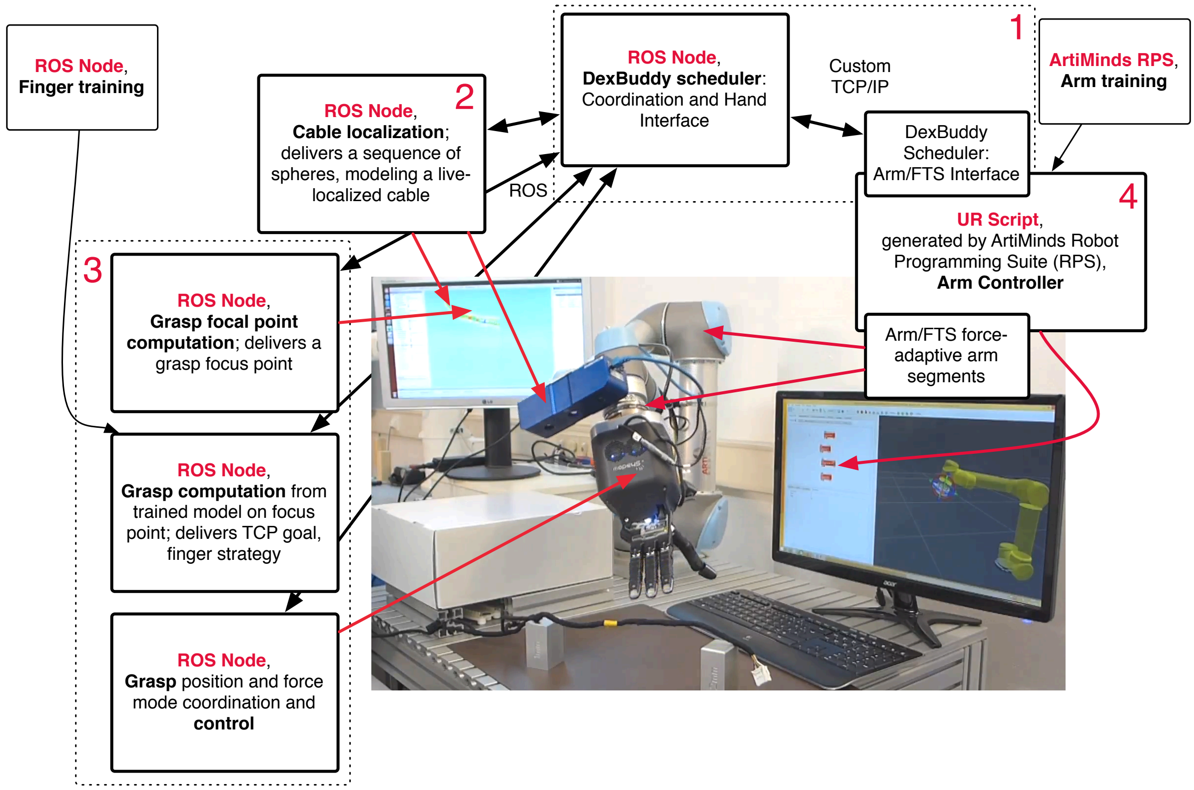
Overview

The DexBuddy software architecture, corresponding software components and specific algorithmic procedures have been developed as planned in the proposal.



Four main groups of modules can be distinguished:

1) The scheduler system, coordinating the abstract task sequence and respective component interaction
2) The cable localization system, using a 30 Hz, industrial point-cloud 3D-camera
3) The grasp computation and execution control system
4) The arm motion and force-control execution system

Components 3) and 4) additionally have a training stage, to easily (re)teach task segments.

Components 2) and 3) are fully based on ROS, while component 4) is fully realized by the commercial ArtiMinds Robot Programming Suite (RPS) and its respective, generated UR Script programs. Component 1) is a hybrid, with the communication between the two platforms realized by a lean TCP/IP interface.

Component 1)

The scheduler is a lean sequence management program, developed to switch between exclusive arm motion and exclusive hand motion segments as stated in the DexBuddy proposal. Its main part is implemented as a ROS node with a sequence visualization and a control GUI for human interaction. On the RPS-side, a special RPS template has been developed exclusively for the DexBuddy project, which automatically generates the UR Script code for the TCP/IP communication interface. The UR Script then actually acts as a server, receiving segment execution commands at arbitrary times.
Practical experience: The coordination of the different components works automatically and well.

Component 2)
The cable localization, developed exclusively for the DexBuddy project, utilizes the Point Cloud Library (PCL) to process the point cloud delivered at up to 30 Hz by the ENSENSO 3D-camera. A complex pipeline of processing filters leads at the end to a chain of spheres, modeling the shape and location of arbitrary cables in arbitrary shape configurations within a certain cable diameter range.
Practical experience: The system is able to localize the cable of the HVAC with a difficult, "woolen" surface and only 8mm diameter surprisingly robustly with a robot-TCP-mounted ENSENSO. It takes around 3 seconds to process at the moment. There is potential for further speed-up.

Component 3)
The hand/finger motion computation and force control module group is the largest portion, developed exclusively for the DexBuddy project. It consists of three layers of hand static/dynamic grasp/manipulation computation and control:
3a): Grasp focal point computation:
The localization system delivers the cable shape and pose as a regular sequence of spheres. The grasp focal point computation module performs several steps based on that input: computing an interpolation/smoothing of that chain of spheres; then computing the area of interest for grasping, depending on the trained grasp (which mostly means relative to the end of the cable); and finally computing a cylinder from neighboring spheres, the center of which is then the grasp focal point.
[Further work, involving deep learning to deliver advanced fingertip pose computation also considering strong, local cable bending is currently still under work, but was not promised in the proposal.]
3b): Grasp adaptation from trained grasp and execution monitoring:
A trained grasp type, matching the selection of the scheduler is loaded from the database and adapted to the computed focal point. The resulting, necessary robot-TCP-pose is computed for the motion planning goal of component 4).
[Further work utilizing the deep learning, mentioned above, as well as MoveIt! for finger motion planning towards the computed fingertip poses is also under way, but was not promised in the proposal].
3c): Low level position / force control management:
The low-level control module manages the position-control motion along the computed finger motion as well as the correct timing for the switch into force control of the fingers and the execution and monitoring of the final force-controlled motion part.
Practical experience:
Grasp computation, based on real, live cable localization data works well as does the switch from position to force control. However, the cable shape around the grasp focal point may not be bent too much. To account even for such extreme cases, a high-end system combining deep-learning neural networks with classical geometric motion planning for the fingers, is currently developed.

Component 4)
The arm motion computation and force control component is mostly comprised of the commercial ArtiMinds RPS. Additional to the part of component 1), the specific arm motion and force control segments are trained using the ArtiMinds RPS and its generated UR Script code.

Practical experience: The system currently lacks one specific motion template type for a certain type of constraint free move. This is the last functionality lacking in the overall DexBuddy software system and will be implemented by ArtiMinds at the beginning of 2016, to be usable for the main experiments.