3D simulation-based user interfaces for a highly-reconfigurable industrial assembly cell

Christian Schlette, Eric Guiffo Kaigom, Daniel Losch
Georgij Grinshpun, Markus Emde, Ralf Waspe, Nils Wantia and Jürgen Roßmann

Abstract— **Although SMEs would benefit from robotic solutions in assembly, the required invests and efforts for their implementation are often too risky and costly for them. Here, the Horizon 2020 project "ReconCell" aims at developing a new type of highy-reconfigurable multi-robot assembly cell which adresses the particular needs of SMEs. At the Institute for Man-Machine Interaction (MMI), we are developing 3D simulation-based user interfaces for ReconCell as the central technology to enable the fast, easy and safe programming of the system.**

ReconCell heavily builds on previous developments that are transferred from research and prepared for industrial partners with real use cases and demands. Thus, in this contribution, we describe MMI's software platform that will be the basis of the desired user interfaces for robot simulation and control, assembly simulation and execution, Visual Programming and sensor simulation.

I. INTRODUCTION

The Horizon 2020 Innovative Action "ReconCell"[1] develops a new type of robot workcell and according user interfaces for automated robot assembly at SMEs. Many SMEs in Europe would benefit from robotic automation, but often refrain from its implementation due to the required invests and efforts to be able to cope with the complexity of setup and maintenance. Robotic automation is, thus, normally economically infeasible for SMEs, especially for small batch sizes. Here, ReconCell develops a multi-robot assembly cell which is fast, easy and safe to (re-)configure and (re-)program. By substantially reducing setup and maintenance efforts, the project aims at commercial and operational viability for SMEs even at batch sizes of about 1000 units.

The key feature of the ReconCell system will be its highly-modular, manufacturer-independent multi-robot design (see Fig. 1) which allows for self-reconfiguration (e.g. by automatically positioning fixtures and sensors). Despite its complexity, (re-)configuration and (re-)programming will be enabled by a comprehensive, functional 3D virtual model of the system, which supports simulation, automated simulation-based optimization as well as simulation-based control of the assembly cell. This central user interfaces will be implemented based on previous developments of the project partners, particularly the VEROSIM® system by the Institute for Man-Machine Interaction (MMI) (see. Sec. II).

All authors are with the Institute for Man-Machine Interaction (MMI), RWTH Aachen University, 52074 Aachen, Germany {schlette, kaigom, losch, grinsphun, emde, waspe, wantia, rossmann}@mmi.rwth-aachen.de

[1]http://www.blue-ocean-robotics.com/en/rd/production/reconcell

At MMI, we work on according technologies as part of the "eRobotics" methodology [1], a development platform for roboticist to exchange ideas and to collaborate with experts from other disciplines. The central method in eRobotics are "Virtual Testbeds", where complex technical systems and their interaction with prospective working environments are first designed, programmed, controlled and optimized in 3D simulation, before commissioning the real system.

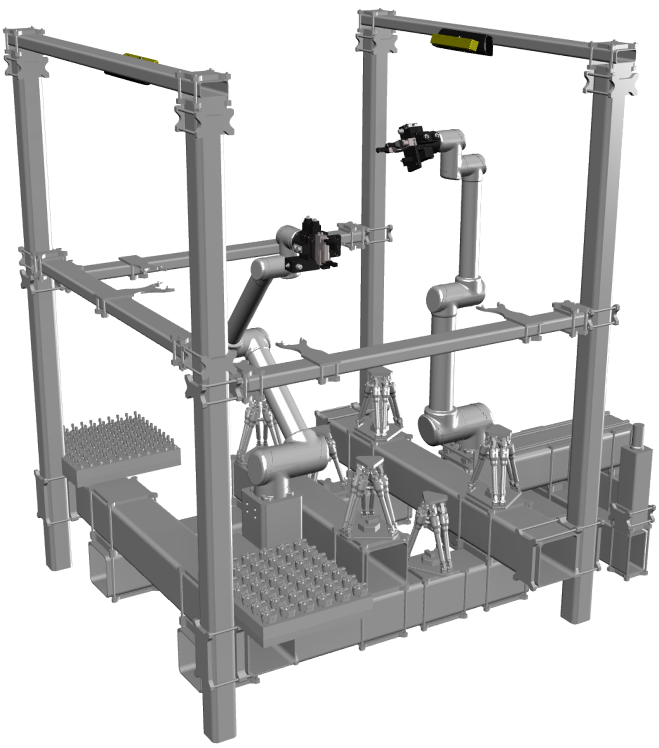

Fig. 1. Design study of the ReconCell system, consisting of two manipulators (here two Universal Robots UR10 equipped with gripper systems by Schunk), passive hexapod fixtures which are reconfigurable by the robots and a frame for cameras and other sensors (here two combinations of Point Grey Bumblebees and Microsoft Kinects) which are also reachable and reconfigurable by the robots.

Fig. 2 depicts the intended workflow with the ReconCell system, which evolves around the functional 3D virtual model of the ReconCell system. The CAD model and additional information about a desired product are given into the 3D simulation-based user interfaces for automated preparations, e.g. geometries and materials for the initialization of collision detection and dynamics simulation as well as a basic description of the desired assembly sequence. The

978-1-5090-1314-2/16/$31.00 2016 IEEE

basic assembly sequence will then be refined to a detailed sequence of applied and parameterized skills mainly by means of visual programming (see Sec. V). The resulting representation based on ”ActionBlocks” will allow for 3D simulation of the assembly process, including simulation of robot kinematics and dynamics (see Sec. III), simulation of assembly steps (see Sec. IV) and sensor simulation (see Sec. VI). The simulation will also enable automated means of optimization, e.g. by deriving optimized positions for fixtures and sensors. Finally, if all steps have been verified in simulation, the very same ”ActionBlock” representation will be the basis for commanding and excuting the assembly process on the real setup.

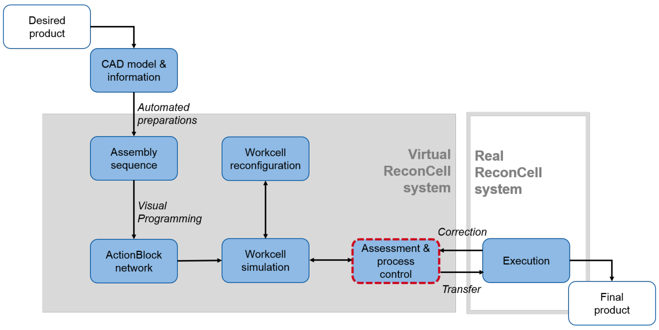

Fig. 2. Intended workflow of the ReconCell system (left to right).

II. SOFTWARE PLATFORM

For realizing eRobotics concepts, the major prerequisite on the tool level is the use of one single but comprehensive and integrated 3D simulation framework which is able to implement the methods and support the processes outlined above.

A. Requirements

The major advantage of such an integrated framework is the ability to simulate all components within one single but comprehensive Virtual Testbed while minimizing conversion tasks between various subsystems. This leads to various requirements of the underlying 3D simulation framework:

- Overall Flexibility: The simulation system must support a broad range of applications and usage scenarios (see Fig. 3), by the way enabling to be used a) as an engineering tool on the desktop, b) as an interactive system to realize complex user interfaces and c) as a tool for realizing control algorithms on real-time capable systems. Hence, it has to separate simulation algorithms from user interface implementation.
- Freely Configurable Database: In order to be able to address the manifold of assembly scenarios at SMEs, the underlying data model has to be freely adaptable to new components and products. To allow all methods to be based on the same model which contains (on an equal level) geometric information as well as e.g. sensor configurations or controller programs, a meta data system and a reflection API are necessary,
- Calibrated Simulation Algorithms: In order to obtain valid and reliable results, the simulation algorithms have to be calibrated against real systems.
- Seamless Transition from Simulation to Reality: The use of block-oriented data models and subsequent code generation for controller implementation is standard in ”Rapid Control Prototyping. This well-established workflow should also be available in the 3D simulation system, such that data processing algorithms developed with and integrated into the system can be used on the real hardware.

The state of the art in simulation technology lists various general approaches to simulation: Discrete event simulation systems [2], block-oriented simulation approaches like the Matlab/Simulink framework [3] or the ”Modelica” modeling language [4] as well as various FEM-based simulation tools (e.g. COMSOL [5]) are probably the most well-known ones. Regarding quasi-continuous 3D simulation technologies in robotics and automation, available approaches are development frameworks (e.g. ROS [6] and GAZEBO [7]) or generic mechatronic systems (e.g. Simscape [8]).

In summary, most approaches focus on dedicated application areas, dedicated disciplines (electronics, mechanics, electronics, thermodynamics, etc.), or are restricted to the development of single components. Still missing is a holistic and encompassing approach, which enables and encourages the synergetic use of simulation methods on a single database throughout the entire lifecycles of technical system.

B. Simulation System Architecture

To overcome these limitations and to fulfill the requirements listed above, we developed a new architecture for simulation systems called VEROSIM®. The key idea is the introduction of a micro kernel, the Versatile Simulation Database (VSD, see Fig. 3). Basically, the VSD is an object-oriented real-time database holding a description of the underlying simulation model. Fully implemented in C++, it provides the central building blocks for data management, meta information, communication, persistence and user interaction. The VSD is called ”active” as it is not simply a static data container, but also contains the algorithms and interfaces to manipulate the data. Furthermore, the VSD incorporates an intelligent messaging system that informs interested listeners of data creation, change and deletion events.

The functionality of the micro kernel is extended by various plugins implementing simulation or data processing algorithms, interfaces to hard- or software systems, user interfaces, etc. Using the VSD, the plugins can communicate with the database as well as establish directed communications between themselves.

III. ROBOT SIMULATION AND CONTROL

A salient example for the practical usefulness of the VSD is its high affinity for a fast development of challenging automation solutions. This development is made possible by the integration capabilities emphasized in Fig. 3 and messaging advantages pointed out in Sec. II-B which, when

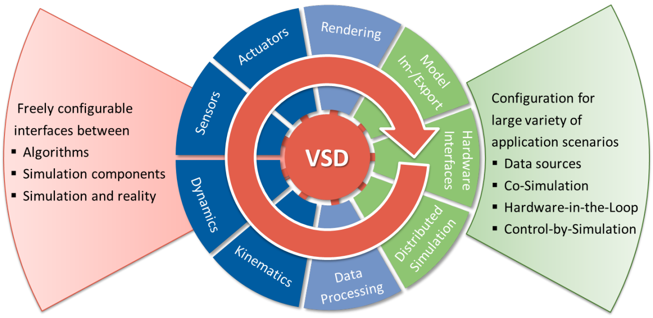

Fig. 3. The VEROSIM® microkernel architecture.

combined, offer ideal tools upon which ready-to-use robotics solutions can be provided. Toward this end, basic robotics functionalities are intuitively put at disposal for a seamless integration into and successful employment in multidisciplinary higher level applications in Virtual Testbeds. Each of these functionalities captures and reflects a fundamental level of abstraction of the behavior of a physical robot in association with intermittent interactions with its workspace. While the kinematic abstraction focuses on a geometric description of the behavior of objects to which a robot and its surrounding environment belong, the dynamic abstraction considers the inertia properties of these objects. Both abstractions are loosely coupled through a third abstraction that delivers control forces to the dynamics in order to meet desired motions specified by the kinematics at runtime.

A. Kinematics

Kinematics in VEROSIM® allow for defining and programming the motion of individual objects as well as kinematic chains, solely based on their position and orientation, velocity and acceleration. Kinematics are often used as a first step to model robots by defining joints between articulated links. In some applications, series of joints can be controlled as individual paths in kinematic trees along trajectories in configuration space or Cartesian coordinates [9]. In the ReconCell context, these trajectories will be the desired joint motions the robot must follow by using additional motion control components for a successful completion of targeted manipulation objectives, as shown in Fig. 4.

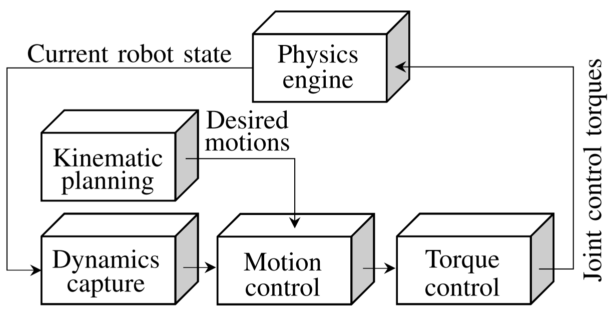

Fig. 4. Simplified architecture of the overall approach for robot simulation.

B. Dynamics

As an exchange of mechanical work in terms of contact force and velocity between a robot and its operational environment becomes a stringent requirement, considering the inertia properties of objects that populate a scene becomes mandatory. For the sake of generality, a uniform treatment of these interactive dynamics has been adopted in Cartesian coordinates. While these coordinates facilitate the modeling of challenging configurations of single bodies that form a robot as well as its workspace, constraint forces are injected, when required, between pairs of bodies for two important goals.

- The first goal is a systematic mechanical assembly of multi-body systems. In the case of a manipulator, this assembly is done by inserting torque-controlled joints between its links for articulation purposes. The same joint functionality also enables a translatory motion of the base of a robot along a linear axis, as is the case with a kinematic abstraction of the same multi-body systems. The choice of the underlying abstraction level is left to the user and will be influenced by the objectives of the task at hand in the ReconCell system.
- The second goal is the rendering of a natural interactive dynamic behavior of multi-body systems. More specifically, a frictional contact dynamics is enforced between each pair of physically interacting bodies by employing constraint forces which prevent these bodies to penetrate each other.

In order to determine the next state of multi-body systems constituting a scene, the physics engine detailed in [10] computes constraint forces by solving a linear complementary problem that characterizes the constraints at hand. These forces are then introduced into an expression of the law of Newton that apprehends the scene as a set on single independent bodies. The Cartesian velocities of the center of mass of each bodies in the scene follow from this expression. Given this set of velocities, the configuration of a particular robot manipulator is easily obtained through an integration of those Cartesian body velocities related to its links.

It is worth to note that at this stage of pure multi-body simulation, a robot modeled in the simulator is not controlled. The robot will sink down under gravity load unless joint control torques are suitably provided as shown in Fig. 4.

C. Control

A model-based trajectory following and interaction control provides the necessary tracking accuracy and skillful disturbance response to external forces to any simulated robot [11]. As shown in Fig. 4, the component that captures the current robot dynamics in joint coordinates, receives the current robot state made of the vectors of joint positions and velocities. On this basis, the inertia, Coriolis and gravity disturbances are systematically captured. This nominal dynamics provides an advantageous insight into the robot behavior. Indeed, this dynamics will be beneficial for the construction of advanced performance indexes that uncover the best possible motion efficiency potentials for the robots

(a) Real setup. (b) Virtual setup.

Fig. 5. Real and simulated setup for simulation fidelity evaluation.

employed in ReconCell. Since the far reaching impact of this insight is not restricted to any robot type or application, the unique information flow on the robot dynamics can be interfaced by any high-lever application for specific purposes. These include different forms of supporting decision taking in business analytics as well as the rationalization of automation processes that rely on the usage of intelligent robots.

In fact, the nominal dynamics of the robot facilitates a subsequent development of numerous skillful motion control approaches, such as joint admittance control. In this compliance control scheme, the goal is to absorb the impact energy in the transient phase of contact and render a desired stiffness at steady state. For this, commanded joint control torques are delivered by the torque control component. These torques are finally fed into the physics engines in order to actuate the robot joints, which leads to a new robot state (see Fig. 4).

D. Robot Simulation Validation

A comparison between the simulated and real joint torques has been carried out to evaluate the fidelity of the simulation. For this purpose, identical joint stiffnesses are commanded for the real and simulated robots in joint admittance control mode shown in Fig. 5(a) and 5(b). During physical interactions with the physical robot, measured external joint torques are forwarded to and used as input by the simulated robot. Tracked joint trajectories, real and simulated joint torques obtained during the validation experiment are shown in Fig. 6. As can be seen, real and simulated joint torques match

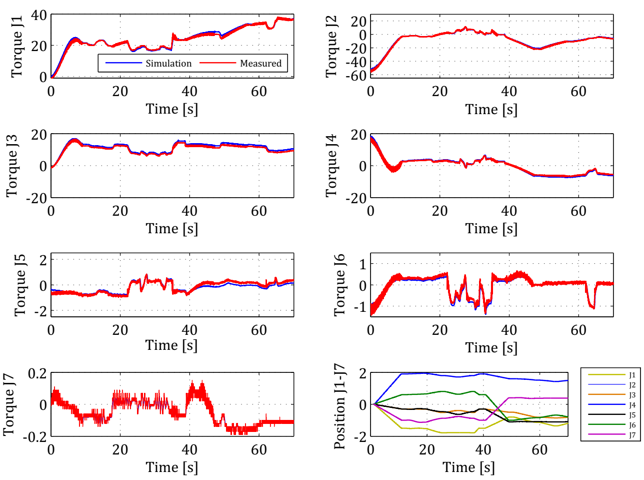

Fig. 6. Comparing real and simulated joint torques.

quite well during free motion (e.g., time $\in [0, 20]_{[s]}$) and physical interactions (e.g., time $\in [20, 40]_{[s]}$).

IV. ASSEMBLY EXECUTION AND SIMULATION

Complex industrial assembly sequences in ReconCell are composed of elementary tasks. Such elementary tasks are enabled in the robot controllers by so-called skill primitives, which represent high-level control functions resp. stategies to solve specific tasks. At MMI, we focus on the development and validation of skill primitives to handle peg and hole operations, since most industrial assemblies can be described in terms of peg and hole connections.

For the development skill primitives to address peg and hole operations, we currently use KUKA LWR robots (see Fig. 5), but will switch to Universal Robots UR10 in future experiments for ReconCell (see Fig. 1). The KUKA lightweight robots are equipped with internal torque sensors sensitive enough to handle contact experiments. In addition, the robots are equipped with SCHUNK force / torque sensors as these sensors provide exact forces and torques at the TCPs and allow for deriving the exact behavior of the parts related to the collision.

During an assembly execution, the robots grasp and bring the peg and hole parts in contact with each other according to strategies described in the skill primitive, while reacting on the situation and and adapting its strategy based on the data from the internal and external sensors. For the simulation of peg and hole operations as part of the ReconCell user interfaces, we model the given assembly scenario in VEROSIM® based on detailed virtual representations of the robots, the sensors and the peg and hole parts. This functional 3D virtual model of the assembly scenario then allows to parameterize, optimize and test specific combinations of assembly components and peg and hole parts using methods of Visual Programming.

V. VISUAL PROGRAMMING

Visual Programming is a programming paradigm that aims to replace or augment conventional, text-based programming. Since visual approaches are in general more concrete, direct, explicit and allowing for direct visual feedback [12], they target inexperienced robo tprogrammers and process engineers utilizing a prototypical development approach. Examples for Visual Programming include the block- and icon-based development of algorithms for LEGO Mindstorms [13] or service robotics [14].

In previous projects we developed the concept of "ActionBlocks" to utilize the Visual Programming paradigm in the development and Virtual Commissioning micro-optical assembly processes [15], [16] and to simulate the results of symbolic planning algorithms that create optimized action sequences and processes [17]. The concept is used to visually program processes for agent-based control architectures: An ActionBlock represents a parameterized action that is executed by an agent – either in simulation or in the physical realm. ActionBlocks can represent actions of arbitrary complexity, from the simple activation of grippers to the execution of move-and-glue sequences that alternate kinematic movements and glue dosage.

Visual Programming with ActionBlocks is primarily carried out in a 2D data flow editor integrated into our simulation system. The data flow editor is used to visually connect ActionBlocks to a sequence of process steps and to assign agents and additional parameters to the ActionBlocks. Fig. 7 shows an excerpt from an assembly sequence that was visually programmed with ActionBlocks.

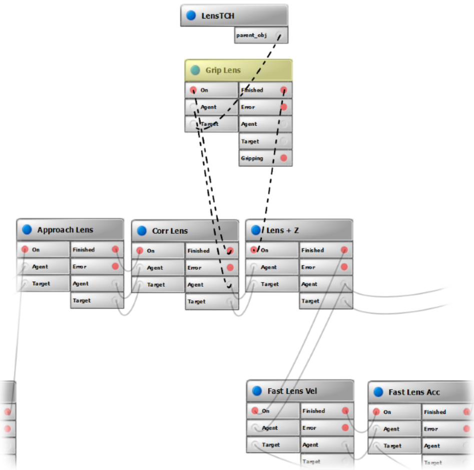

Fig. 7. Example of an ActionBlock network from Visual Programming.

VI. SENSOR SIMULATION

The 3D simulation-based user interfaces in ReconCell depend on a realistic simulation of optical sensors, since only the application of sensors allows to bridge the gap between the ideal, virtual model and the real ReconCell system. The basis of the optical sensor simulation is the sensor framework in VEROSIM® [18]. It provides methods for the modeling, simulation and visualization of a wide range of sensors and offers a consistent data interchange within the simulation environment as well as between real sensors and simulation algorithms in hardware-in-the-loop scenarios. Logging and playback mechanisms allow for an efficient offline development for real sensors while the introduction of various error models enable the detailed analysis of sensor data processing algorithms under different boundary conditions [19].

In detail, the sensor framework supports the parallel integration of real and simulated sensors into the system and provides a smooth transition between simulation and real world setups. The design of the sensor framework allows for easy setups of 3D virtual models using a generic communication concept for the interaction of all components offering a focused view on every component of the system to analyze and optimize its behavior. Fig. 8 shows the sensor framework's system structure.

Due to the fact that simulated sensors are implemented to deliver or work on ideal data, specific error models need to be applied to grant close-to-reality simulations required

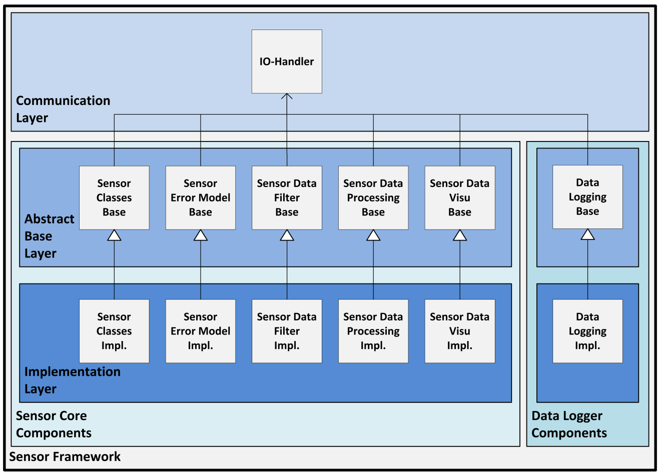

Fig. 8. Structure of the sensor framework.

for Virtual Testbed scenarios. The system allows the realistic simulation of a wide range of optical effects and errors. Electronic and optical effects with a huge impact on computer vision based algorithms are distortion, chromatic aberration, depth-of-field, noise and sensor saturation. On modern, programmable graphics hardware it is possible to map the characteristics of an optical sensor to a camera model of the render engine of the underlying simulation system by using shaders. As sensors like cameras deliver data consisting of several millions of measurement values, it is even more important to move the calculations from the CPU to the GPU to fulfill real-time simulation criteria. In contrast to other approaches, our approach takes place as an additional render pass which mostly utilizes data already loaded to the GPU [20].

Our approach allows multiple noise functions and supports the test and verification of computer vision algorithms. The different noise functions are hot-pixel noise, color noise and monochrome noise [19]. The noise characteristics are taken from real cameras. The simplest form of noise, hot-pixel noise, is obtained by taking images in complete darkness, but different temperatures. These images are used as noise textures and are added to the rendered image in a post-processing step. The reproducibility of highly dynamic noise effects can be accomplished through the active simulation time as seeds for distribution of semi-random noise values.

Distortion values for optical systems are provided by the optics manufacturer or can be measured or computed using standard computer vision algorithms as provided by OpenCV, MatLab Camera Calibration Toolbox or Zemax. Approaches to simulate distortion besides other effects are presented in [21] and [22].

Furthermore, the sensor framework supports sensors to directly gather depth information, for example time of flight (TOF) sensors. These send out a light beam towards the scene, that is being reflected and partly returned to the sensor. The TOF sensor calculates the distance to an object using the light's time of travel, that can simply be calculated. Another method is the measurement of the phase shift. A continuous wave laser continuously emits light with a modulated phase. Thus, the distance of an object can be calculated regarding the phase-shift of the received signal. From a simulation point of view, TOF sensors can be regarded as perfect pinhole cameras like the camera models in real-time rendering. Thus, they can easily be mapped to the graphics hardware. Various error models implemented for camera simulation can be applied to the output image of the simulated TOF sensor. Fig. 9 shows an image gathered by a simulated TOF sensor.

Fig. 9. Image gathered by a simulated TOF sensor in a simplified ReconCell setup. In particular, the simulated TOF sensor allows for the parameterization and optimization of algorithms for object recognition and pose estimation early on in the project.

VII. Conclusion

In conclusion, the ReconCell project is aiming at addressing the needs of SMEs by providing a) a highly reconfigurable multi-robot assembly cell and b) 3D simulation-based user interfaces for a fast, easy and safe setup of assembly processes along a consistent workflow (depicted in Fig.2). This contribution presents previous devlopements at MMI, since ReconCell is a H2020 Innovative Action which heavily build on existing results.

In the final phase of the project, we will demonstrate the capabilities of the ReconCell system for three real use cases provided by the SMEs in our consortium (Elvez doo from Slovenia, Logicdata GmbH from Austria and Precizika Metal from Lithuania) as well more use cases established through an open call. In addition, the consortium has recently started to establish a network of potential ReconCell users to disseminate information about ReconCell technologies to all stakeholders in the value chain and to raise awareness about the possibilities of automated robot assembly in SMEs.

Acknowledgement

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 680431.

References

[1] M. Schluse, C. Schlette, R. Waspe, and J. Roßmann, "Advanced 3d simulation technology for eRobotics," in *Proc. IEEE International Conference on Developments in eSystems Engineering (DeSE)*, 2013, pp. 151–156.

[2] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation (3rd Edition)*. Prentice Hall, 2000.

[3] Matlab Simulink. The MathWorks, Inc. Natick, Massachusetts, United States. [Online]. Available: www.mathworks.de/products/simulink

[4] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Computer Society Pr, 2003.

[5] COMSOL Multiphysics. Comsol Group. Stockholm, Sweden. [Online]. Available: www.comsol.com

[6] Robot Operating System. Open Source Robotics Foundation. [Online]. Available: www.ros.org

[7] GAZEBO. Open Source Robotics Foundation. [Online]. Available: gazebosim.org

[8] Simscape Multibody. The MathWorks, Inc. Natick, Massachusetts, United States. [Online]. Available: www.mathworks.de/products/simmechanics

[9] C. Schlette, *Anthropomorphe Multi-Agentensysteme: Simulation, Analyse und Steuerung*. Springer-Verlag, 2013.

[10] T. Jung, "Methoden der mehrkörperdynamiksimulation als grundlage realitätsnaher virtueller welten," Ph.D. dissertation, RWTH-Aachen University, Aachen, Germany, 2011.

[11] E. G. Kaigom and J. Roßmann, "Developing virtual testbeds for intelligent robot manipulators-an erobotics approach," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1589–1594.

[12] M. Burnett, "Software engineering for visual programming languages," in *Handbook of Software Engineering and Knowledge Engineering*. River Edge, NJ, USA: World Scientific, 2001, vol. 2, pp. 77–92.

[13] S. H. Kim and J. W. Jeon, "Programming LEGO mindstorms NXT with visual programming," in *Proc. International Conference on Control, Automation and Systems (ICCAS)*, 2007, pp. 2468–2472.

[14] C. Datta, C. Jayawardena, I. Kuo, and B. A. MacDonald, "RoboStudio: A visual programming environment for rapid authoring and customization of complex services on a personal service robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2352–2357.

[15] C. Schlette, D. Losch, and J. Roßmann, "A visual programming framework for complex robotic systems in micro-optical assembly," in *Proc. International Symposium on Robotics (ISR/Robotik)*, 2014, pp. 750–755.

[16] C. Schlette, D. Losch, S. Haag, D. Zontar, J. Roßmann, and C. Brecher, "Virtual commissioning of automated micro-optical assembly," in *Proc. SPIE Laser Technology and Industrial Laser Conference, Part of Photonics West (LASE)*, 2015, dx.doi.org/10.1117/12.2080742.

[17] N. Wantia, D. Losch, and J. Roßmann, "Virtual commissioning and symbolic planning of micro-optical assembly processes," in *Proc. Computer Science and Electronic Engineering Conference (CEEC)*, 2015.

[18] M. Emde, J. Roßmann, B. Sondermann, and N. Hempe, "Advanced sensor simulation in virtual testbeds: A cost-efficient way to develop and verify space applications," in *AIAA Space 2011 American Institute of Aeronautics and Astronautics (AIAA) Conference and Exposition, Long Beach, California*. AIAA, 2011, pp. 1–6.

[19] J. Roßmann, N. Hempe, M. Emde, and T. Steil, "A real-time optical sensor simulation framework for development and testing of industrial and mobile robot applications," in *Proceedings of 7th German Conference on Robotics (ROBOTIK 2012)*. VDE, 2012, pp. 1–6.

[20] J. Roßmann, N. Hempe, and M. Emde, "New methods of render-supported sensor simulation in modern real-time vr-simulation systems," in *Proceedings of the* 15^{th} *WSEAS International Conference on Computers*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2011, pp. 358 – 364.

[21] M. Kučiš and P. Zemčík, "Simulation of camera features," in *Proceedings of the* 16^{th} *Central European Seminar on Computer Graphics*. Technical University Wien, 2012, pp. 117 – 123. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php?id=10161

[22] M. Kučiš, P. Zemčík, O. Zendel, and W. Herzner, "Overview of Simulation of Video-Camera Effects for Robotic Systems in R3-COP," in *SAFECOMP 2013 - Workshop DECS (ERCIM/EWICS Workshop on Dependable Embedded and Cyber-physical Systems) of the* 32^{nd} *International Conference on Computer Safety, Reliability and Security*, M. ROY, Ed. Toulouse, France: Springer Science+Business Media B.V., September 2013, pp. 1 – 8. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00848628