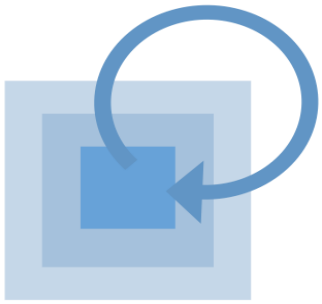

2011

Deliverable Report

D6.3 Web environment hosting simulations, compilers and chem. sensor information (M24)

Matrix for Chemical IT (MATCHIT)

Grant Agreement no. 249032

Deliverable 6.3: Web environment hosting simulations, compilers and chem. sensor information

Web environment: Establishing a web environment that hosts (i) physicochemical- and artificial chemistry simulations, (ii) code for a DNA readdressing language for chemtainers (part of a "MATCHIT compiler"), (iii) code for the real-time chemtainer monitoring and control feedback system, as well as (iv) an update of the technical issues associated with integrating the Chem-IT matrix: physicochemical modelling vs. experiments; data input from instrumentation vs. IT control; IT logical operations vs. DNA computing in the matrix).

This deliverable serves all WP 6 objectives as well as WP 8 (dissemination).

It describes the MatchIT software repository which is part of the MatchIT website, located at http://fp7-matchit.eu/index.php?page=software-repository . Purpose of the online repository is to ease access to all scientific software developed within the scope of the project. Each software package is available for download free of charge and – in the majority of the contributions – as open source software.

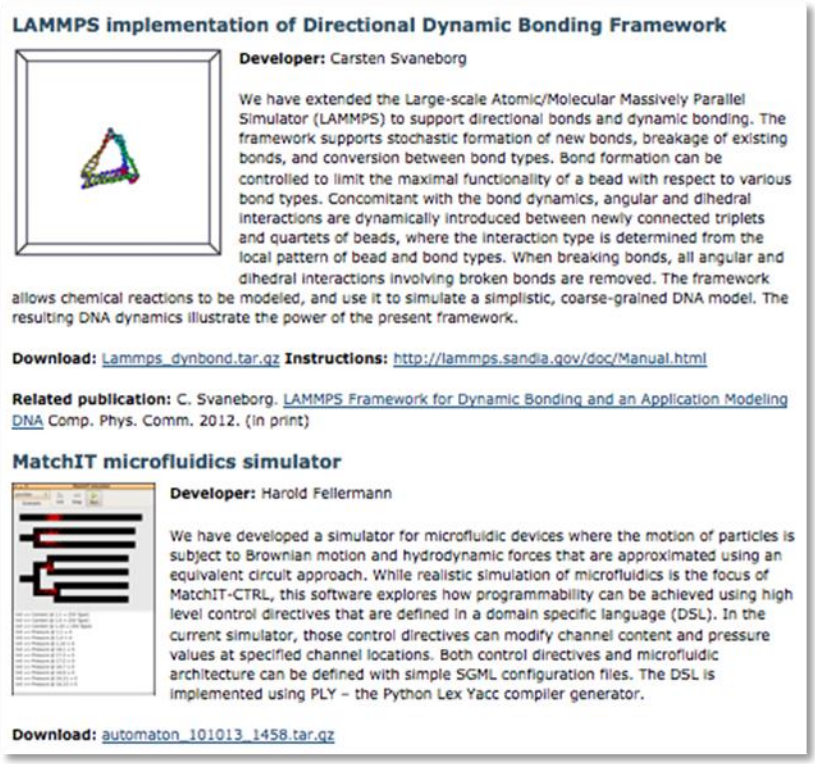

Figure 1: Screenshot showing part of the online software repository for MatchIT.

While the MatchIT software repository is intended to be comprehensive within the next months, it already hosts most of the software development activities of the project, which currently comprises the following list:

LAMMPS implementation of Directional Dynamic Bonding Framework

Developer: Carsten Svaneborg

We have extended the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) to support directional bonds and dynamic bonding. The framework supports stochastic formation of new bonds, breakage of existing bonds, and conversion between bond types. Bond formation can be controlled to limit the maximal functionality of a bead with respect to various bond types. Concomitant with the bond dynamics, angular and dihedral interactions are dynamically introduced between newly connected triplets and quartets of beads, where the interaction type is determined from the local pattern of bead and bond types. When breaking bonds, all angular and dihedral interactions involving broken bonds are removed. The framework allows chemical reactions to be modeled, and use it to simulate a simplistic, coarse-grained DNA model. The resulting DNA dynamics illustrate the power of the present framework.

Ionic-Chemistry Library

Developer: Abishek Sharma

The ionic-chemistry library calculates the time and spatial dependence of the concentrations of ionic species depending on the time-dependent electrode voltages and the nonlinearly coupled local electrostatic potential, via an extended Poisson-Nernst-Planck framework, which takes finite ion size effects into account. From this, the forces to which the tracer particles in the simulation are subjected can be directly derived (proportional to the gradient of the potential). The treatment is fully nonlinear, and treats finite size effects via an entropic excluded volume correction that is important, because otherwise ions can pile up to unphysical concentrations at oppositely charged electrodes. The procedure is made efficient by mapping the 3D geometry to a 1D framework, extending theory developed recently for the PNP equations in membrane ion channels. Novel here is the transformation of coordinates derived for a general area function that allows a non-singular description of the general pseudo-1 D problem generalizing the case of hemispherical electrodes to electrodes embedded in a channel. We use the solution of Laplace's equation to derive the static 3D shape of potential curves and then use the full time dependent solution to assign potential values to different positions. This captures the main physical effects of the exceedingly complex interplay between electronic and migration and reaction behavior in aqueous solutions. The Mathematica solution structure must then be exported to C and coupled with sparse linear matrix equation solver libraries for integration with the ng_biopro software. This is aided by Mathematica's inbuilt C export capability, but requires some additional matching with external libraries.

DNA Address Compiler

Developer: Weizmann Institute (P3b)

In the DNA address compiler and design checker, tags are generated by using an evolutionary algorithm: an initial group of ssDNA sequences are randomized. For each pair of primers a Needleman–Wunsch global sequence alignment score is computed. It is assumed that alignment scores correlate with the tendency of primers to form undesirable dimers in solution. The pair with the highest score is chosen and mutated. Mutations are carried out by substituting a random base in one of the pair with a different base and re-computing alignment score. The first mutation found to lower alignment score is selected and preserved. Next, all scores are recomputed and again the pair with the highest score is mutated. The whole process is repeated for new groups of sequences until

no further improvement is achieved over several hundred cycles, for each group. The group of sequences with the lowest sum of all scores is provided as an address library.

MatchIT microfluidics simulator

Developer: Harold Fellermann

We have developed a simulator for microfluidic devices where the motion of particles is subject to Brownian motion and hydrodynamic forces that are approximated using an equivalent circuit approach. While realistic simulation of microfluidics is the focus of MatchIT-CTRL, this software explores how programmability can be achieved using high level control directives that are defined in a domain specific language (DSL). In the current simulator, those control directives can modify channel content and pressure values at specified channel locations. Both control directives and microfluidic architecture can be defined with simple SGML configuration files. The DSL is implemented using PLY – the Python Lex Yacc compiler generator.

Real-time simulation in MATCHIT-CTRL

Developer: Uwe Tangen

The purpose of this simulation-facility is twofold. On the one hand it should be used to test and debug electrode-activation patterns and state-machines because debugging them in a real experiment can be very time consuming. On the other hand the simulation must be fast enough to be able to run in parallel with the experiment to allow an on-line comparison between simulation and experiment. This feature should be used to extract parameters from the experiment which are otherwise not seen. Here simulation is used as a world-model in the experiment.