The purpose of a domain model is to concisely capture the concepts of an application's domain, and their relation among each other. Even though the main purpose of domain models is not on implementing the application, major parts of an application can be generated from the application's domain models fully automatically with today's technologies. The focus of today's code generation technologies, however, is mostly on the structural aspects of the domain; the domain's behaviour is often not modelled at all, or implemented manually based on some informal models, or the behaviour is modelled on a much more technical level.

The Event Coordination Notation (ECNO) allows modelling the behaviour of an application on a high level of abstraction that is closer to the application's domain than to the software realizing it. Still, these models contain all necessary details for actually executing the models and for generating code from them. In order to be able to model the behaviour of a domain, the ECNO makes the events in which the different elements of the domain could engage explicit. The local behaviour of an element defines at which time an element can engage or participate in an event. The global behaviour of the application results from different elements jointly engaging in such events, which is called an interaction. Which events are supposed to be jointly executed and which elements need to join in is defined by so-called coordination diagrams of the ECNO. Together, the models for the local and the global behaviour define the overall behaviour of the domain.

In this technical report, we discuss the main idea and philosophy of ECNO and its notation as well as all the subtle details and concepts - and we motivate the decisions made for its design. Moreover, we discuss the prototypical implementation of ECNO, which consists of a modelling environment based on Eclipse and the Eclipse Modeling Framework (EMF) and an execution engine, which fully supports all the concepts and features of ECNO discussed in this technical report. All the examples are based on EMF, but the ECNO Engine can be used with different other platforms or object-oriented code across different platforms, once some adapters are provided. Though the focus of this technical report is on the general concepts of ECNO, the examples discussed here work for version 0.3.2 of the ECNO Tool and Framework. The ECNO Tool as well as the examples are available from the ECNO Home page: http://www2.imm.dtu.dk/~ekki/projects/ECNO/.