The MPI standard is complex and defines several usage patterns as erroneous.

Thus, MPI programming errors are common and often difficult to detect, which has given rise to a variety of MPI correctness tools. One particular focus of these tools is deadlock detection, which is more complicated than traditional deadlock detection since it includes potential deadlocks, which only occur for some MPI implementations or certain configurations. Furthermore the standard allows calls that wait for one out of multiple messages, this leads to a need for more general deadlock models. Umpire is the first tool with a precise deadlock detection strategy. However, this approach comes with several performance and correctness problems. This thesis presents a theory for deadlock detection in MPI applications, its implications for correctness tools, and a prototype implementation is added to the Umpire framework. To understand MPI deadlocks, enhanced wait-for graphs (WFG) are used. Since a cycle in these graphs is not a sufficient MPI deadlock criterion, this thesis introduces a novel criterion, called OR-Knot. This also enables visualization of MPI deadlocks.