Will Harwood

I did not have slides but this is what I said:

i) I recently complete a Ph. D. thesis at York entitled "The Logic of Trust" - it addresses the question of how we use our social embeddedness to answer questions about trust. The work is an attempt to synthesise viewpoints from computation and social theory and, in particular, develops two questions relating to trust and their computational solutions:

a) "Knowledge on Trust": What is "the best"state of knowledge that we can obtain from information sources of mixed trustworthiness that may contradict one another;

b) "Social Trust": What is the best trust assessment we can have of another individual based on the social network within which we are embedded and the conflicting information we have about the trustworthiness of individuals in this network.

The primary tools used in the formal modelling are relational calculus, Galois connections and boolean model checking.

The thesis will shortly be available at the White Rose Repository: http://eprints.whiterose.ac.uk or by request to the author wth501@york.ac.uk

The second item is an announcement of work in progress which is a joint venture with Jeremy Jacobs the University of York.

ii) Information flow channels in systems can be understood as binary relations between (collections of) inputs and (collections of) outputs. Predicates over inputs and outputs can be understood as sets of inputs and sets of outputs and general information flows can be understood as relations between such sets.That is, we may a general information flow as connection predicates on inputs to predicates on outputs (integrity), and predicates on outputs to predicates on inputs (confidentiality). These flows correspond to weakest precondition induced by the binary relation, and the weakest precondition induced by its converse, in a natural way:

Let wp.R.Y = weakest precondition of Y through binary relation (program) R and conv(R) = converse of R

wp.R.Y = X => providing an input x in X guarantees an out in Y

wp.conv(R).X = Y => seeing an output y in Y guarantees the input was in X

The forthcoming paper explores these ideas in detail and relates them to standard ideas on program information flow.