Motivation and problem statement

The problem: A software system is typically composed of a variety of artefacts (e.g. source code, UML diagrams, etc.). In practice, different artefacts evolve at different rates and modifications applied to one artefact may not get reflected in other related artefacts.

Consequences of differential evolution and outdated artefacts:
- Synchronisation issues
- Inconsistency among artefacts
- Lack of trust in artefacts by stakeholders
- Impediments to effective system maintenance and evolution

Although incremental development provides a more flexible solution for handling changes [1], artefact consistency is often neglected; different representations of software go through stages of refinement without all the dependent artefacts being considered as the process does not enforce artefact links.

State of the art – Evaluation

Conclusions:
- Partial solutions in terms of artefacts supported
- Certain aspects of artefact consistency are supported in isolation [2][3][4]

Figure 1 – classification of solution space

Artefact consistency management solution space

Holistic view of artefact consistency

An ideal consistency management solution supports all of the following activities:
- Traceability creation & maintenance
- Change detection
- Impact analysis
- Consistency checking
- Change propagation

Prototype system

Functionality:
1. Extract fine-grained information on elements and their relationships from original artefacts & store this information in an interim XML format
2) Save elements and their relationships in a graph database (Neo4j) [5]
3. Detect changes affecting artefacts
4. Perform impact analysis

Selected artefacts:
Requirements documents,
UML class diagrams, source code

Future work
- Complete prototype implementation & evaluation
- Some challenges to be addressed: distributed software development, scalability, ensuring that the prototype is non-intrusive and does not enforce the use of particular methodologies and tools.

References:

Figure 2 – Processes supported by the prototype

Figure 3 – Graph representation