WG1: 人間中心的なソフトウェア開発

ソシオテクニカル情報空間としてのソフトウェア

葉　雲文@(株)SRA

ヘンリーフォードの組み立てラインのような生産方式は，ソフトウェア業界にとって長年の夢であり，最終の目標として追求してきたため，工場メタフォーは，これまでのソフトウェア業界の考え方を支配してきた．しかし，ソフトウェアは車のような物理的なモノではないため，ソフトウェアを作るプロセスも生産するプロセスと捉えるべきではないと考える．

プログラムの語源はpro (before) + graphein (write)であり，write publiclyという意味合いで，planやscheduleの意味をもっている．つまり，プログラムは生産される物理的なものではなく，書かれる知識アーティファクトである．ソフトウェア開発は知識創出プロセスとみなすべきである．そのプロセスの唯一の資源となるのは情報とプログラマが持つ知識である．

大きいソフトウェアシステムを開発にあたって，多数の開発者が携わらなければならない．これも工場生産と違って，開発者の人手(multiple hands)を求めているだけではなく，複数の頭脳(multiple heads)が持っている知識の複合的な共創効果を求めなければならない．

ソフトウェア開発が知的共創作業であるという視点から，ソフトウェア開発組織は，ひとつの知識ネットワークを生成するとみなすことができる．この知識ネットワークには，二種類の知識ノードが存在する．第一は，知識が組み込まれたアーティファクトとしてのプログラムファイルやドキュメントファイル，第二は，知識を持っているプログラマである．開発というプロセスは，開発者が持っている知識をアーティファクトに組み込むプロセスであり，知識が開発者からアーティファクトに外在化されるプロセスであると考えられる．また開発プロセスは，他の開発者の作ったアーティファクトを利用したり参照したりすることにより，そのアーティファクトに組み込まれた知識を獲得するプロセスでもある．つまり，知識がアーティファクトから開発者に内在化されるプロセスである．開発者は，プログラムに関する知識を完全に外在化することはできず，開発者が直接なコミュニケーションを通して暗黙知を交換することが，開発時においてきわめて重要であることが報告されている．Robillardらの研究によると，ソフトウェア開発者がプロジェクト開発に関する暗黙知を交換するためのコミュニケーションに費やした時間は開発時間の41%をも占める．これらの知識交換が必要となるのは，開発者が抱えている開発タスク間に，多様な依存性が存在するためであると考えられる．

その多様な依存性は，単なるタスク間のテクニカル的な依存性だけではなく，開発者間のソーシャル的な関係も含むので，ソフトウェアプロジェクトが織り出した知識ネットワークを一つのソシオテクニカル情報空間(STIS)とモデルし，開発にあたって必要とする知識をよりやすく獲得するような仕組みの提供に関して研究開発を進めている．

下図はソシオテクニカル情報空間(STIS)を示す．STISには開発者，コード，そして文書という三つのノードがある．各ノードを繋ぐ実線は，それらのノード間のprimary relationを表す．これらの関連は，ソフトウェアシステムと開発履歴の解析から直接計算した関連である．たとえば，コードの構造上の依存関係，開発者がどの部品を作ったか，どの文書を作成したかなどはprimary relationである．破線はcomposite relationを表す，つまり，primary relationの組み合わせにより生まれた関連である．Composite relationを利用して，開発者の時々刻々と変換する情報ニーズに合わせて，適時に知識を獲得する経路を構成することを目指している．本会議ではいくつかのcomposite relationとその効用を紹介する．