

図3 KAOSの記述例

3. シューティングゲームの記述

ゲームでは、ある明確なゴールが存在しており、プレイヤーがその達成を目指すため、ゲームの要求定義にゴール指向でのアプローチは有効であろうと考えた。ゲームとしては、今回は、シューティングゲーム[6-7]を対象とすることにした。シューティングゲームでは、固定された画面の中で、画面外からランダムに出現する敵キャラクタに対してプレイヤーが弾を発射して破壊する。このゲームの目的は、一次的には敵を倒すことと生き残ることである。爽快感や満足感については、一次的な目的を満了した時に達成される二次的な目的として考えた。研究においては、このゲームを NFR, i*, KAOS により、記述と分析を試みた。

(記述は付録参照)

4. 比較結果と考察

三つの表記法では、著しく記述方法が異なる場合がある。例えば、ソフトゴールの1つである「クリアできること」について比較すると、NFR では、クリアのための条件を次第に詳細化していくことでそれに必要な操作を明らかにしていく。i*では、勝利や敗北するためにはどのアクタが他のアクタに関わるか、を記述していく。KAOSでは、クリアというゴール達成のために必要となるソフトゴール、オブジェクト、オペレーションなどに分解していき、最終的にどのエージェントがそれを達成するのかを明確にしておくことになる。三つの表記法を比較しまとめたものを表1に示す。また、グラフとして比較すると、NFRとKAOSは木構造なのに対して、i*はネットワーク構造である。記号の種類については、NFRでは3種類、i*では5種類、

表1 記述の定性的評価

NFR	<ul style="list-style-type: none"> 解決すべきソフトゴールは分かるが、ゴール達成のための機能が不明確 各要素の横のつながりの関係性がわかりにくい。 上位ゴールとの関係の理解が容易 木構造のためグラフ的に単純
i*	<ul style="list-style-type: none"> 依存関係の記述による繋がりを表現できる。 アクタ間及びアクタ内部の依存関係をそれぞれ記述し分析できる。 受益者と提供者の関係が分かり、各アクタの働きが表せる。 ネットワーク構造のためグラフ的に複雑
KAOS	<ul style="list-style-type: none"> エージェントの記述により責務関係がわかる。 ゴール達成のための複数のエージェントの連携が明確 木構造（エージェントの繋がりを除く）のためグラフ的に単純

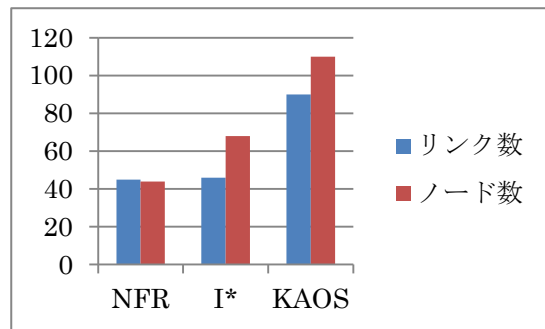


図4 ノード数とリンク数の比較

KAOSでは6種類である。シューティングゲームでの記述について（ノード数、リンク数）を比較すると、NFR (45, 44), i* (46, 68), KAOS (90, 110) となった。(図4参照)

一般的に、記号の種類が少ないほどシンプルで理解容易であるが、表現力は乏しくなり、逆に、記号数が多いほど表現力は増すが、習得に時間がかかる。また、ネットワーク構造の方が木構造に比べ、表現力に柔軟性があるが、グラフが複雑になると理解が容易ではなくなる。但し、木構造についても、下位に同じ要素のノードが複数個離れて出てくる可能性があり、グラフの規模が大きくなると理解が容易とは言えない場合がある。

以上をまとめてみると、NFRでは、記号が3種類と少なくシンプルであり、木構造を用いて詳細化するため構造が

分りやすいが、表現力は低い。i*では記号が5種類とNFRに比べ多く、ネットワーク構造のためにグラフとして複雑な構造になり易いが、より詳細な表現が可能である。KAOSでは、記号数が6種類と最も多く表現力は最も高いが、習得までに時間がかかる。KAOSについてはノード記号数が最も多い上に、シューティングゲームの記述結果である図4を見ると、リンク数とノード数も最も多いという結果になっている。一般的に、記号数が多く表現力が高い方が、グラフはコンパクトになりそうだが、逆の結果となった。この理由は、KAOSではエージェントまで記述するが、NFRとi*では記述しないということと、記号数が多いために表現する内容が増え、またそれらの記号間に関係が生じリンクが張られるためである。今回、シューティングゲームを記述した結果、個々のノードや構造全体の表現力、エージェントの責務の明確化という点から考えて、分析に最も適していたのはKAOSであった。

5. おわりに

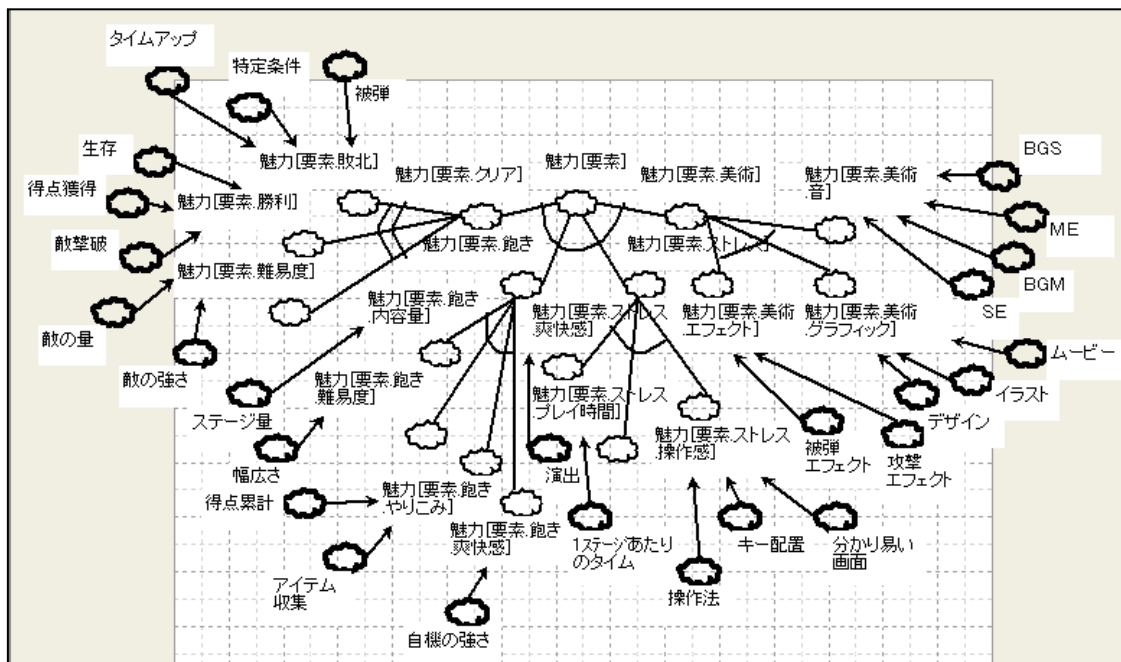
本研究において、ゴール指向手法は、ゲームアプリケーション開発でゴールに関連する要素の関連を明確にする上で有用であることが分かった。シューティングゲームを三つの代表的なゴール指向手法であるNFR, i*, KAOSで比較を行うと、KAOSを使用するのが最も適していることが

解った。これは、アクション、格闘、RPGゲーム等、シューティングゲームと似たような特徴を持つゲームにおいても同様と考えられる。なお、本稿での対象はシングルプレイヤーゲームであったので、プレイヤー間の利害関係の衝突や協調などが生じるマルチプレイヤーゲームで、引き続き研究を行っている。

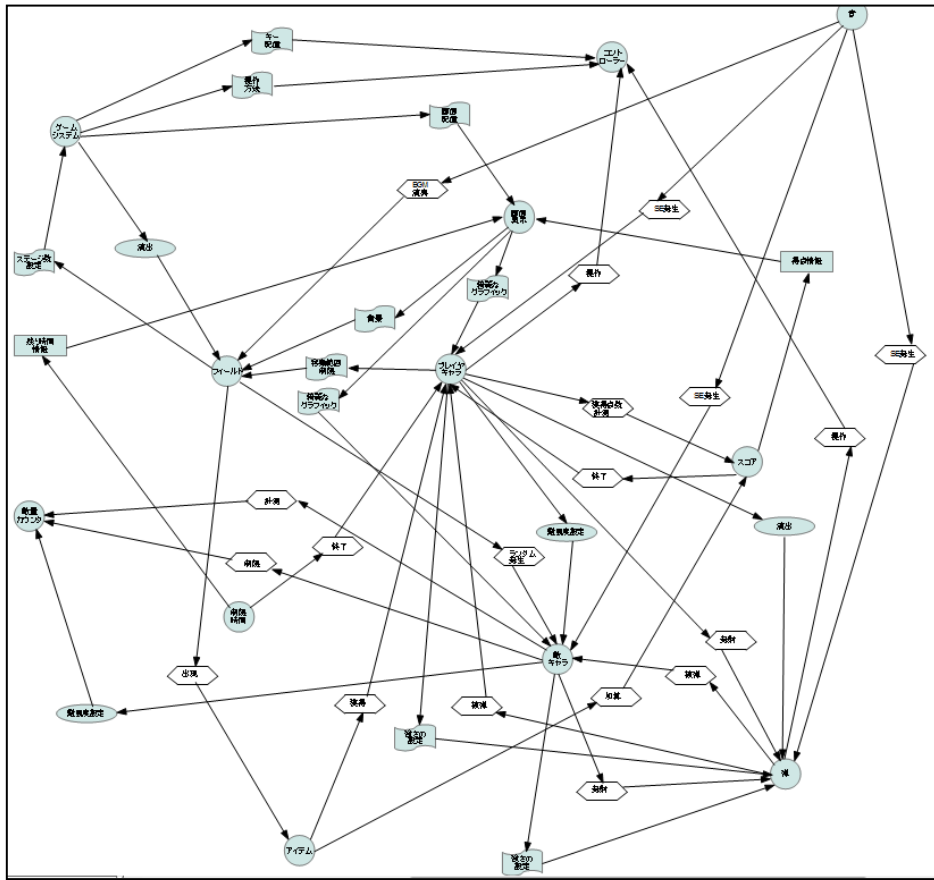
参考文献

[1]山本修一郎:ゴール指向によるシステム要求管理技法, ソフト・リサーチ・センター(2007).
 [2]山本修一郎:非機能要求とゴール指向要求定義, 情報処理学会誌, Vol.49, No.4, pp371-379(2008).
 [3]L. Chung, B.A. Nixon, E. Yu, J. Mylopoulos: Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers (1999).
 [4]i*Webpage:http://www.cs.toronto.edu/km/istar/
 [5]A. Dardenne, A.van Lamsweerde, S. Fickas: Goal-directed Requirements Acquisition, Science of Computer Programmig, Elsevier, Vol.20, pp.3-50(1993).
 [6]中嶋謙互:オンラインゲームを支える技術, 技術評論社(2011).
 [7]松浦健一郎, 司ゆき:シューティングゲームプログラミング, ソフトバンククリエイティブ(2008).

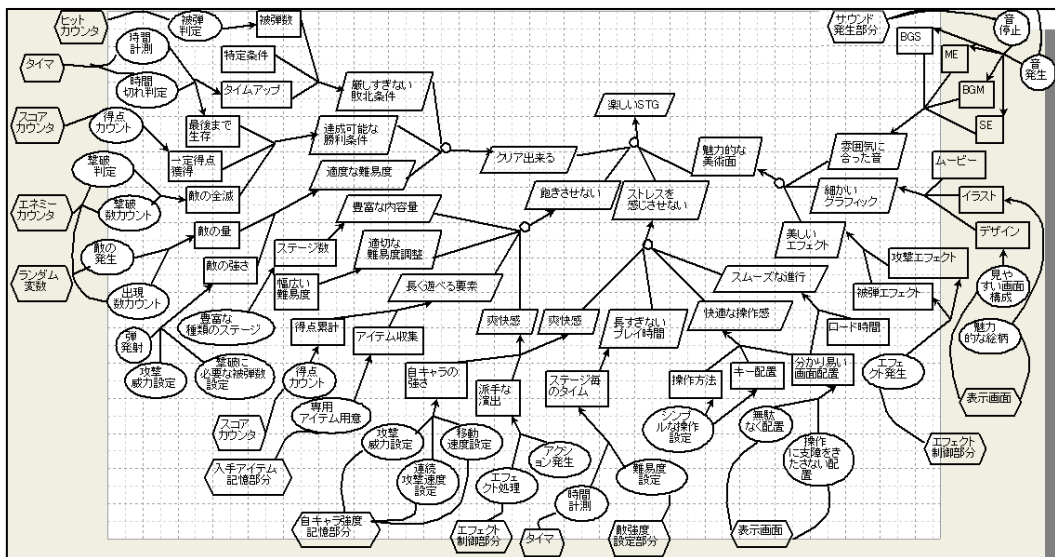
付録 三つの手法によるシューティングゲームの記述・分析結果



図A1 NFR フレームワーク



図A2 i*フレームワーク



図A3 KAOS