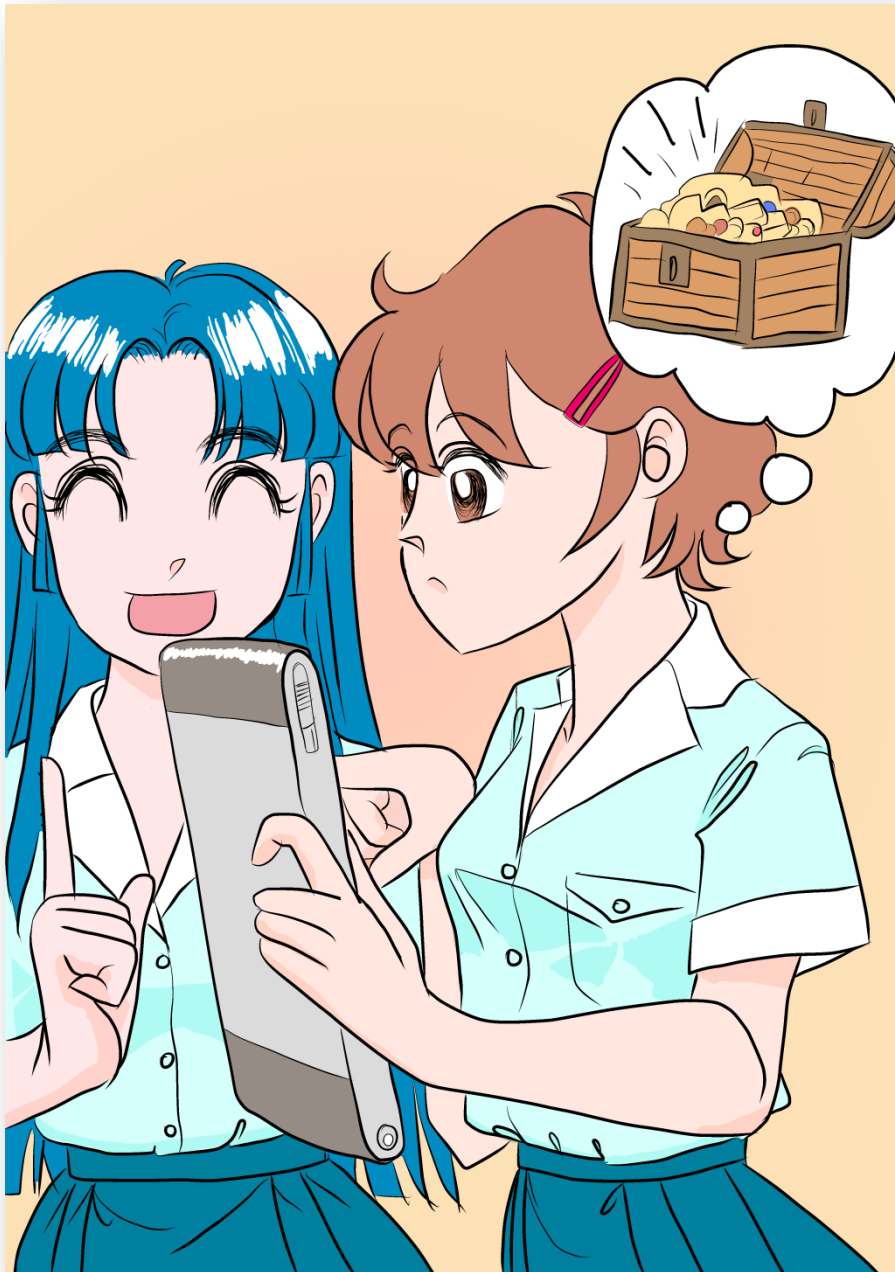


XGMTK 概要



プロジェクト Web サイト : <http://xgmtk.org/>

プロジェクト連絡先アドレス : develop@xgmtk.org

文責 : 椎路ちひろ (犬山屋算譜)

(本文書は 2011 年 08 月 13 日に XGMTK report No.1 に掲載したものからの抜粋です)

目次

1 XGMTK 開発の動機と目的	4
1-1 TRPG プレイへの社会的制約の影響	4
1-2 コンピュータ RPG ～ MMORPG	5
1-3 XGMTK の目的と目標	7
2 XGMTK の概要	10
2-1 遠隔セッション支援機能	11
XMPP の採用	11
ルールに基づく対話の制御	12
GM ベースのセッション	13
キャラクタ会話とプレイヤー会話の分離	13
プレイヤー、GM、キャラクタの識別	13
プレイヤーMUC	14
キャラクタ MUC	14
ルール駆動対話と行動結果通知	15
ダイス	16
2-2 データ共有支援機能	16
XML の採用	16
データ形式と GUI の分離	17
回線障害・チート対策とデータ更新	17
汎用性	18
単位系	19
ゲーム内時間	19
イベント	19

空間.....	21
エリア・タイプ.....	22
キャラクタやアイテム.....	24
秘密の管理.....	24
2-3 自動化支援機能.....	25
スクリプト言語の実装.....	26
2-4 オーサリング支援機能.....	26
3 開発目標と方針、計画.....	27

XGMTK は現代的なコンピュータ技術、ネットワーク技術を取り込んだ TRPG のプレイ環境構築のためのソフトウェアで、XGMTK プロジェクトは XGMTK を開発することを目的とするプロジェクトです。本文書では XGMTK の開発を企画するに至った動機、XGMTK の概要と要素技術、開発予定のソフトウェアに求められる要件と大まかな設計案について説明します。

1 XGMTK開発の動機と目的

まずは XGMTK の開発を企画するに至った動機、そして XGMTK の目的について説明します。

1-1 TRPG プレイへの社会的制約の影響

筆者がTRPG¹を知ったのは1980年代後半、大学に入ったばかりの頃でした。それから大学院修了までの約6年弱、筆者はTRPG、主に「D&D」、「Rune Quest」と「クトゥールの呼び声」といったゲーム²で多くの休日やゲーム仲間と楽しくセッション³して過ごしたものでした。特に後二者は筆者自身もGM⁴として、オリジナルのシナリオ⁵を作るなどして大いにハマっておりました。

しかし学生の頃には友達と気軽に遊んでいた TRPG も就職で地元を離れると縁が切れてしまい遊びにくくなってしまいました。職に就く場合被雇用者・給与所得者となる人は多く、企業の本社が小数の都市部、特に日本では東京と大阪に集中する傾向から、このようなことは恐らく筆者だけの経験ではなく、首都圏をはじめとする大都市圏以外に生まれ育った日本人には必ずしも珍しくない経験なのではないかと思われます。(大学も大都市圏に集中する傾向がありますので就職以前に進学で同じような目に会う人も多いことでしょう。)

さらに企業は気楽に社員に転勤を命じますし、転職は昔に比べて増えたと言っても流石に居住地のコミュニティに残る目的で転勤を避けるために転職というのも一般に簡単ではありません。このため折角就職後に新天地で TRPG 仲間 (趣味仲間というのも居住地に作り上げられるコミュニティの一つと断言していいでしょう) を作っても長く維持するのが難しい場合も多くなってしまいます。TRPG では、キャンペーンで長く遊ぶことにも大きな魅力がありますから TRPG 仲間と長く付き合い続けられない社会状況は大きな障害だと言えます。これはプレイヤー間の「距離」の問題と言えます。

上述のように就職をきっかけに人が都市部へ移動するために都市部ではメンバーを集めやすくなる効果が

¹ テーブル・トーク・ロール・プレイング・ゲームの略。机を囲み紙と鉛筆を媒介にして、ゲーム内の人物 (キャラクター) として役割演技 (ロール・プレイング) を行い協力的に空想世界での冒険や探索といった目的を達成するゲーム

² 他には Phantasm Adventure や WARPS、トラベラーなど

³ TRPG をプレイする集まり一回分のこと。

⁴ ゲーム・マスターの略。プレイヤーにゲーム内でキャラクターが置かれた状況を説明する進行役であり、行為の成否を判定するレフェリーでもある役回り。

⁵ ゲームでプレイヤーが演じるキャラクター (プレイヤー・キャラクター、PC) が置かれた背景や状況、場所、達成すべき目的、制約条件、解決すべき障害などを設定したもの内、特定のセッションのために準備された固有部分。プレイヤーに対しての秘密を含むため一般に GM が管理する。それ以外の一般的な部分はルールとして GM とプレイヤー間で共有される。

あります。しかしその一方で、都市部に居住するプレイヤーは都市部の住宅事情の悪さの影響を受けます。都市部では何人かで集まってセッションが可能な部屋のある家は必ずしも多くはないという問題があるからです。特に就職のために都会へ出てきて間もない単身者の居住環境は悪く、決して家賃が安いわけでもないワンルームマンションでも居室は非常に狭く、数人が集まるのも難しいような場合が珍しくありません。TRPGでまとまったセッションを行おうと思うと泊りがけということも多いわけですが公的な場所で宿泊が可能で安く気軽に集まれる場所というのも中々厳しいものがあります。(TRPGとは関係ないですが、この日本の状況を思うと、コンビニの前が深夜に行き場のない若者のたまり場になるのも仕方ない結果なのかもしれません。)これでは就職直後にTRPGから離れる人が増えるのは仕方ないことでしょう。これはプレイに必要な「空間」の問題と言えます。

また被雇用者・給与所得者となる人が多いことから、学生時代に比べ勤務時間の拘束がきつくなります。特に日本は有休消化率が低く、勤務先によっては週休2日すら完全実施できていないような長時間労働の国ですからTRPGセッションのために集まる時間を作るのも一苦勞、ましてGMを引き受けてオリジナルのシナリオを用意するなどということは時間的にかなりの難易度ということになります。これはプレイに必要な「時間」の問題と言えます。

以上のように、現代の社会状況はTRPG趣味に大きな影響を与えており、以下のような形で持続的なプレイの障害になっているのではないかと思います：

- 距離
- 空間
- 時間

TRPGが学生の遊びの域を超えて大人でも十分楽しめる生涯の趣味となるにはこの障害を克服ないし緩和することが必要でしょう。

1-2 コンピュータRPG ~ MMORPG

前項で書いたような事情もあってか、RPGと言えばTRPGよりもむしろコンピュータRPGが思い起こされるというくらいに普及しています。ではコンピュータRPGはTRPGの代わりになれるのでしょうか？

初期のコンピュータRPGはTRPGで言うところのソロプレイ⁶やゲームブック⁷の延長線上にあるものでした。このためGMも一人のプレイヤーが操作するPC⁸以外のNPC⁹もコンピュータ・プログラムであり、柔軟性

⁶ 一人でGMとプレイヤーを兼ねる一人遊びTRPG。GMしながらでも秘密が分らないようにシナリオ作者がシナリオの記述にゲームブック的な工夫をすることで可能になる。

⁷ コンピュータ・ゲームにおけるアドベンチャー・ゲームの原型。ユーザの選択やサイコロの目などの結果によって物語が分岐し、1作品内で複数の物語が展開できるように書かれた本。

⁸ プレイヤー・キャラクタの略。プレイヤーが演じるキャラクタ。物語でいえば主役。コンピュータRPGではアバターと呼ばれるゲーム内の分身としてCGで表現されることも多い。

に欠け、仲間とのセッションのといった楽しみのないものであり、しばしば作業プレイに陥り易いものでした。コンピュータ・ゲームに対する習熟がもたらす作業プレイそのものは、ハマると表現されるような、一種の瞑想のような没入感をもたらすという快樂もあるのでそれはそれで立派に一つの楽しみということは言えるでしょう。とはいえ、それは明らかにTRPGとは異なる種類の楽しみと言えます。

近年、インターネットの普及によりマルチユーザ・オンライン・RPG (MORPG) が一般的なものとなりました。マルチユーザ・オンライン RPG では数人の人間が同時にログインして各キャラクタを操作することができるコンピュータ RPG で、仲間とのセッションが可能になりました。これはもちろん画期的な前進ではありましたが、相変わらず GM は柔軟性に欠けるコンピュータ・プログラムであり、NPC はロボットのように決まり切ったことしかできない状況には変化がありませんでした。

しかしここ数年、通信回線の高速大容量化、サーバの能力増大により数百人～数千人規模の同時ログインを可能とするマッシブ・マルチユーザ・オンラインRPG (MMORPG) が稼働するようになりました。相変わらずGMはプログラム¹⁰でNPCはロボットの様であるとはいえ、MMORPGではNPCに対して圧倒的多数のPCが存在し、小規模な社会といえるものを形成できるため、ゲーム内経験におけるNPCの比重が下がり結果としてより多彩なプレイ経験が味わえるようになりました。とはいえ、現在のMMORPGでは1つのワールド・サーバへの同時ログイン数はせいぜい数千の規模であり、村程度の規模の社会しか実現できませんし、PCは冒険者などに職業が偏る傾向があり、またプレイヤーは常時いるものでもないので、PCだけで世界を構成してシミュレーション的に運営するようなことはもちろん不可能です。従って一見世界を丸ごと再現したかのような錯覚を覚えるMMORPGにおいても、まだかなりの部分をプログラムされた柔軟性のないGMやNPCに頼る必要があります。

MORPG、MMORPG も含め、コンピュータ RPG は確かに TRPG にはない優れた性質もあります。例えばコンピュータ RPG はコンピュータ技術により動画、3D 表現、音楽、リアルタイムの戦闘といったリッチなマルチメディア素材による多彩な表現を提供できるようになりました。さらに MMORPG では TRPG では事実上不可能だった百人～千人という圧倒的な多人数同時プレイも可能になりました。

しかしその一方でコンピュータRPGの開発にはコンピュータ技術に関する知識と長い開発期間が必要であり、特にリッチなマルチメディア素材の作成やMMORPGの運営は事実上個人では不可能な規模に達していません¹¹。TRPGではシナリオ製作側となって様々なたくらみを張り巡らすことも重要な楽しみの一つでしたが、コンピュータRPGでは一般にユーザはプレイヤーの立場に置かれることが多く、TRPGでは気軽に入れ替わり

⁹ ノン・プレイヤー・キャラクタの略。プレイヤーが演じる訳ではないキャラクタ。物語でいえば脇役、敵役。プレイヤーが知らない秘密の情報を握っている場合もあるため TRPG では GM が管理することが多い。

¹⁰ MMORPG ではオンラインでユーザ向けのトラブル相談窓口を提供する仕組み用意されその担当者が「GM」と呼ばれることが多いが、彼らの権限は一般に限られたものであり、ゲームのレフェリーでも進行役でもないため伝統的な TRPG における GM とは役割がかなり異なる。MMORPG でも伝統的な TRPG における GM の役割である進行役とレフェリー役はコンピュータ・プログラムが担っていると言える。

¹¹ 「RPG ツクール」などコンピュータ RPG 用フレームワークとオーサリング・ツールや、「ダンジョンシージ」のように MORPG でシナリオを作成できるエディタ、「セカンドライフ」などに見られるアバター作成ツールといったような、コンピュータ RPG そのものあるいはシナリオや素材等の開発コストを下げ、ユーザが製作に参加できるようにする試みはある。

が可能だったシナリオやルールを生産者と消費者の立場はくっきり分かれてしまうことになりました。

以上から現代コンピュータ RPG は TRPG を完全に代替するものとはいえないでしょう。しかし比較して得失をまとめると以下になるでしょう：

- 長所 1：リッチなマルチメディア素材の利用
- 長所 2：ネットワークを介した遠隔地の複数プレイヤーの同時プレイの実現
- 短所 1：プログラムによる柔軟性に欠けた機械的なゲーム運び、NPC
- 短所 2：製作・運営コストの増大による製作と消費者の分離

これらから長所を取り入れ、短所を除くことでより望ましい TRPG プレイ環境を構築することが可能であると考えられます。

1-3 XGMTK の目的と目標

このように MMORPG まで含めてコンピュータ RPG は伝統的な TRPG にはなかった長所を備える一方で、生身の GM や NPC が持つ臨機応変な柔軟さ、ローカル・ルールやシナリオ作成時の敷居の低さといった TRPG の長所については当面解決できる見込みはないと考えられます。ですがコンピュータ RPG で示されたコンピュータの長所を伝統的な TRPG に取り入れることは可能であり、それは冒頭で述べた TRPG のプレイヤーに対する距離、空間、時間といった障害の緩和に役立ち、また TRPG のプレイ経験をリッチにするものと考えられます。XGMTK プロジェクトは以上を実現するためのソフトウェアとして XGMTK (eXtensible Game-Master Toolkit) を開発することを目的とするプロジェクトです。

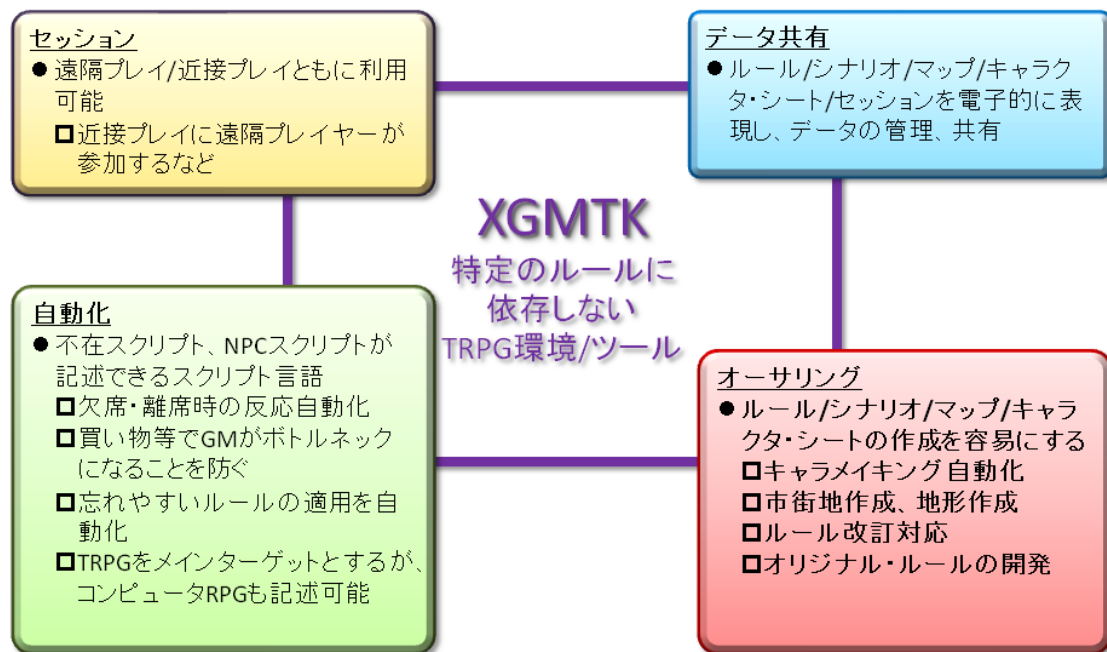


図 1:XGMTK 概要

XGMTK は特定の TRPG 向けのソフトウェアではなく、それらに共通のインフラ、基盤を提供するソフトウェアです。XGMTK では大きく分けて以下の機能を提供します：

- 遠隔セッション支援機能
- データ共有支援機能
- 自動化支援機能
- オーサリング支援機能

以下、各機能について簡単に説明します：

遠隔セッション支援機能は MORPG、MMORPG で有用性が証明されたネットワーク技術の応用となります。インターネットを介し遠隔地にいる複数の人々が一つのセッションで TRPG を遊ぶことを支援し、TRPG に対する障害の内、空間と距離の問題を解決します。またプレイヤーが集合する条件を緩和することで短時間のセッションを手軽に開催できる効果が期待できるため、時間の問題についても幾らか緩和されることが期待できます。詳細は「遠隔セッション支援機能」(p.11) で述べます。

データ共有支援機能はコンピュータ RPG で当たり前提供されているマルチメディア素材の提供を実現します。またキャラクター、マップ、イベントやアイテム、シナリオやルールといった伝統的な TRPG のデータや規則を電子化し、それらの情報のよりきめ細かな管理を実現します。さらに電子化された素材やシナリオは流通も容易になり、ユーザ間での素材やシナリオの交換・共有を促進できるでしょう。またこの機能はそれらのデータを操作する洗練されたユーザ・インターフェースも提供することで TRPG のプレイし易さを向上し

ます。プレイし易さの向上はルールの学習に費やす時間を減らすことでプレイヤーの時間の問題を緩和することでしょう。詳細は「データ共有支援機能」(p.16)で述べます。

自動化支援機能はスクリプト言語による煩雑な作業の自動化や不在時の自動応答を支援します。XGMTKの主要な目的はTRPGプレイの支援環境であり、基本的に人間がGMを務め、ゲームを進行し、行為の成否を判定することを前提としています。とはいえTRPGには自動化しても支障のない部分、自動化によって物事がスムーズになる部分というのがあります、それらの自動化によってGMの負担を減らすことができます。また自動化機能によりプレイヤーの欠席や離席をフォローしてTRPGプレイにおける時間の問題を緩和することも期待できます。さらに、プロジェクト本来の目的ではありませんが、自動化支援機能によりソロプレイ・シナリオの記述という形でコンピュータRPGの開発を行うことも可能になるでしょう。詳細は「自動化支援機能」(p.25)で述べます。

オーサリング支援機能はデータ共有支援を実現する各種マルチメディア素材やキャラ作成、マップ製作、シナリオ製作、ルール記述自動化支援を実現するスクリプトの開発といったような様々なものの開発のためのツール群を提供します。これによってGMによるセッションの準備にかかる時間を短縮し、TRPGプレイにおける時間の問題を緩和します。また、ユーザが素材やシナリオ、ルールの作成側に回することを容易にし、GM＝開発者とプレイヤー＝消費者という役割の固定化を解消し、入れ替わり可能なものとします。これによって素材やシナリオの開発が促進されることも期待されます。詳細は「オーサリング支援機能」(p.26)で述べます。

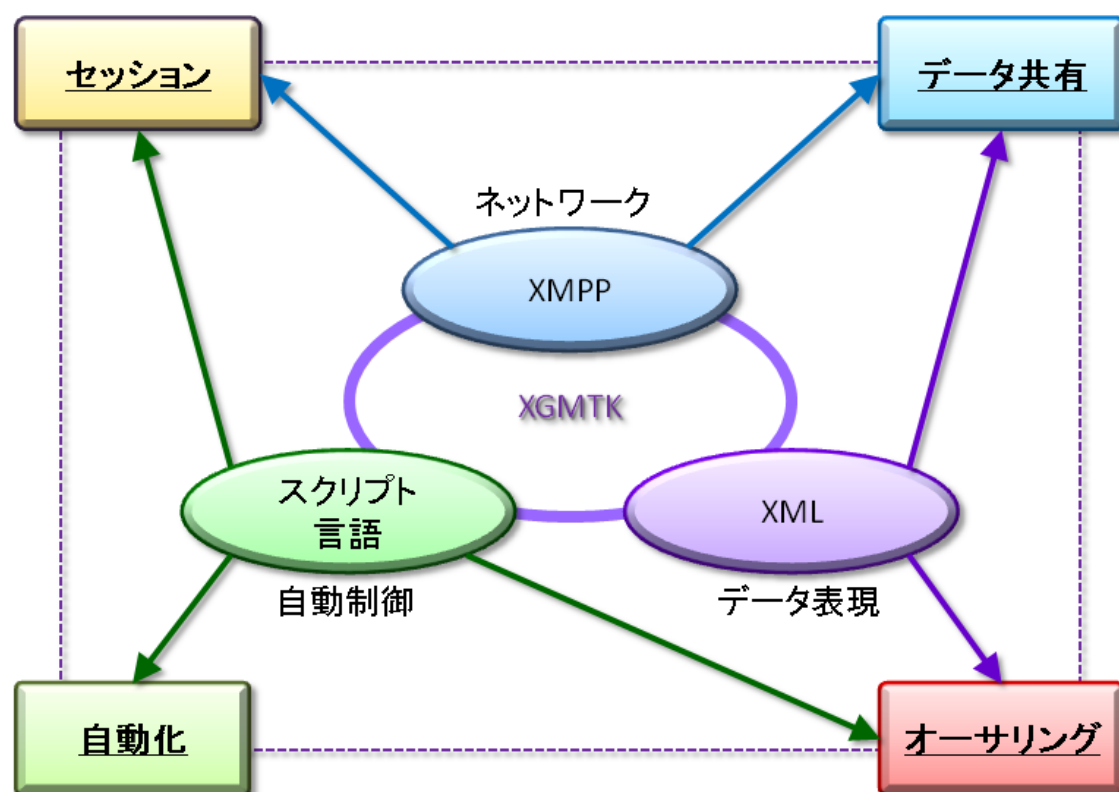


図 2 : XGMTK と情報技術

XGMTK ではこれらの機能を実現するために以下の 3 つの情報技術を採用します：

- XMPP
- XML
- スクリプト言語

XMPP は通信プロトコルの規格であり、主に遠隔セッション支援機能の実現に利用します。詳細は「遠隔セッション支援機能」の「XMPP の採用」(p.11) で述べます。

XML はデータの表現形式の規格であり、データ共有支援機能とオーサリング支援機能の実現に利用します。詳細は「データ共有支援機能」の「XML の採用」(p.16) で述べます。

スクリプト言語は主に自動化支援機能の実現のために利用しますが、ルールやシナリオの記述にも利用しますので、データ共有支援機能とオーサリング支援機能にも関係しています。詳細は定まっていますが、どのような仕組みが必要であるかについては「自動化支援機能」(p.25) で述べます。

2 XGMTK の概要

XGMTK の各機能をどのように実現するかについて概要とそこで利用される要素的な技術について説明します。

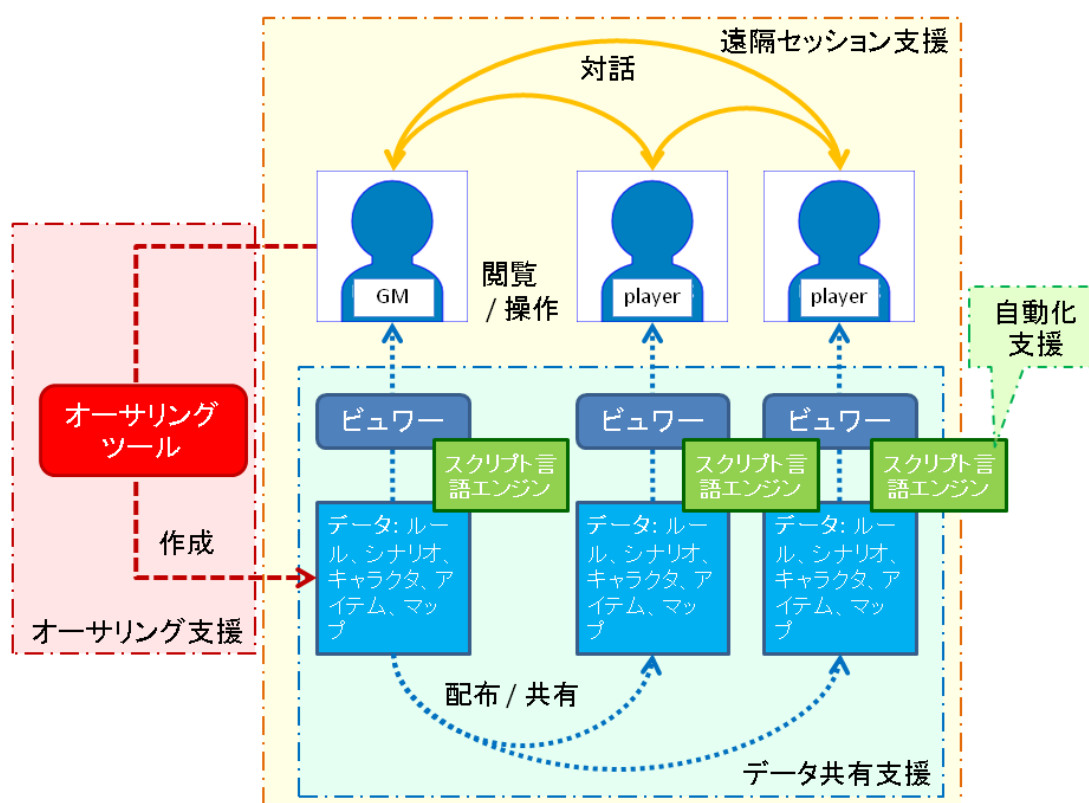


図 3 : XGMTK の構成

2-1 遠隔セッション支援機能

具体的にはインスタント・メッセージング¹²を拡張し、GM・プレイヤー間、プレイヤー相互間の会話、ゲーム進行、レフェリングとデータ共有を可能にすることでそれを実現します。遠隔セッション支援はあくまでTRPGの支援を主眼とするもので、基本的に人間のGMの存在が仮定されるところが通常の本MORPGとは異なります。

XMPP の採用

XGMTKでは遠隔セッション支援のためにインスタント・メッセージングを拡張する必要があります。そのためインスタント・メッセージング・プロトコルとしてXMPP¹³という規格を採用します。XMPPはeXtensible Messaging and Presence Protocolの略でインスタント・メッセージングについてのオープンな通信プロトコルです。その特徴はプロトコルがXML（「データ共有支援機能」の「XMLの採用」(p.16)で詳述)を利用して拡張可能、かつ厳密に定義されていることです。またJava、Python、PHP、C、C++、C#等複数の開発言

¹² MSN Live MessengerやSkype、GoogleTalkのように短いメッセージをリアルタイムでやり取りできるソフトウェア。近年では音声や動画による会話やファイル共有などもサポートしている。

¹³ 規格公式ページ(英語) : <http://xmpp.org/>

語向けにライブラリ、オープンソースのクライアント・ソフトウェアやサーバ・ソフトウェアも整備されています。

XGMTK ではまずはテキスト・チャットを基本としますが、XMPP は音声、映像チャットの規格も整備されているため将来的にはそのように拡張していくことも計画しています。

ルールに基づく対話の制御

現状、一般のインスタント・メッセージングを利用した TRPG の遠隔プレイは既に行われています。ではなぜ改めて TRPG 用のソフトウェアを用意する必要があるのでしょうか？それは一般のインスタント・メッセージングはあくまでチャットおよびファイル共有をサポートするだけで、以下のようなサービスを提供しないためです：

- ルールに基づく対話の制御
- 共有データの管理
- 自動化

共有データの管理と自動化についてはそれぞれ「データ共有支援機能」(p.16)、「自動化支援機能」(p.25)でそれぞれ後述します。ここではまずルールに基づく対話制御について説明します。

TRPG では以下のような対話を区別することが有用だと考えられます：

- プレイヤー間の会話
- キャラクタ間の会話
- 行動の表明
- 行動の結果の通知

例えば、前の二者はプレイヤー、キャラクタにより比較的自由になされるべきですが、後ろの二者はルールの制御の下、順序良く定まった形式で行う必要があります。またプレイヤー間の会話には特に何の制限もありませんが、キャラクタ間の対話はゲーム内の状況により制限される場合があります。例えばキャラクタが地理的に複数に分断されている場合などが挙げられます。

しかし一般のメッセージング・クライアントを用いた遠隔セッションではこれらを区別しないためユーザが意識的に使い分ける必要があります。目前で対話している通常のセッションと異なり遠隔セッションでは対話のチャンネルが限られるためこのような使い分けは時として煩雑なものとなるでしょう。

XGMTK は対話を分類し、ルールに基づいてソフトウェア的にサポートすることで TRPG セッションのスムーズな進行を支援します。

GM ベースのセッション

XGMTK では基本的な対話の単位として数人のプレイヤーが 1 人の GM と行うセッションを採用します。この対話単位は人間の GM を前提にしていますが、コンピュータ RPG で言えば MORPG に近いと言えるでしょう。これは TRPG が個人や小グループのレベルでの冒険や探索、戦闘を主要なターゲットとするゲームとして発展してきた歴史を反映しています。実際問題としてプレイヤーが 1 人からせいぜい数人のキャラクターの挙動をロール・プレイすることを考えるとその視点から見えるゲーム内の風景はせいぜいその程度の規模であるでしょうからこの前提は RPG に向いていると言えます。

MMORPG に見られるような大都市や戦場といえども個人に認識できるのはその一部でしかなく、あとは遠景・背景と考えてよいだろうと思われまますので、そのような場合については、複数の GM により複数のセッション間で連携を取る形が考えられます。とはいえそれは将来の課題とし、当面は一つのセッションで閉じた形を前提にする予定です。

キャラクター会話とプレイヤー会話の分離

XGMTK のセッションにおいてプレイヤー間の会話とキャラクター間の会話はマルチユーザーチャットとなります。インスタントメッセージングにおける通常の対話は 1 対 1 であり、複数人と会話をしていても各対話は独立に扱われます。一方、マルチユーザーチャットは複数の人間が入り混じって会話を行うことができます。

XGMTK のセッション以下の 2 つのマルチユーザーチャット (MUC) からなります：

- プレイヤー MUC
- キャラクター MUC

これら 2 つのチャットはシステム的には区別されますが、プレイヤーや GM に対して実際にどのように表示するか、どのように操作可能とするかはクライアント・ソフトウェアに任せられます。

プレイヤー、GM、キャラクターの識別

XGMTK で採用する XMPP においては各ユーザに Jabber-ID という識別子が振られます。Jabber-ID は一般に「ユーザ名@サービス名/リソース名」という構造をしています。このうち「サービス名」は一般にサーバ名になります。「ユーザ名@サービス名」というメールアドレスそっくりの部分は Bare-JID (ベア JID、裸の JID) と呼ばれユーザを一意に区別できます。「ユーザ名@サービス名/リソース名」は Full-JID (フル JID、完全な JID) と呼ばれ、同一ユーザが複数の場所あるいは複数のクライアントでログインしているときにそれらを区別できます。

XGMTK では GM およびプレイヤーに各一つの Bare-JID を割り当てます。そして各 PC にはそれを操るプレイヤー (各 NPC の場合はそれを操る GM) の Bare-ID 後ろにキャラクターを表す「リソース名」をつけた Full-JID を割り当てます。

プレイヤーMUC

プレイヤーMUCはGMとプレイヤー全員によるMUCです。これは全く通常のMUCであり、ゲームに関する打ち合わせや離席の連絡、食事休憩の提案などセッションを開催、継続するための連絡、およびゲームに無関係なちょっとした雑談等に利用されます。

キャラクタMUC

キャラクタMUCはGMとキャラクタ全員によるMUCです。キャラクタにはPCのみならずNPCがMUCの参加者として含まれます。このMUCはゲーム内でキャラクタ同士の会話とGMによる状況説明を表現するために利用されます。キャラクタの会話は実際にはプレイヤー(PCの場合)なりGM(NPCの場合)が行うわけですが、その内容はキャラクタが発話したものと扱われることになります。どのようなGUIでそれを実現するかはクライアント・ソフトウェア次第ですが、一人のプレイヤーやGMが各々複数のキャラクタを担当できるようなGUIを実現する必要があります。キャラクタMUCはプレイヤーMUCと異なり以下に示すようなゲームの進行、シナリオまたはルールによる幾つかの影響を受けます：

- キャラクタ間の距離と声や音の大きさ
- キャラクタの使用言語
- 会話量とゲーム内時間経過の連動
- 「かくかくしかじか」機能

キャラクタMUCではキャラクタ(PCとNPC)の位置情報を管理しており、キャラクタ間の距離や向きに応じて会話や音の聞こえ方が変化します。特に小さな声や音、遠くの声や音は聞こえにくく、逆に大きな声や音、近くの声や音は聞こえやすいように扱われます。中でも特に小さな音や特に遠くの音は全く聞こえません。またGMによる状況説明などもどの地点の説明かが管理され遠くにいるキャラには伝わりにくく、あるいは伝わらないように扱われます。これによって分断されたパーティや、ヒソヒソ話、叫び声、聞き耳を立てるといったような状況を表現することができます。

キャラクタMUCではルールとシナリオの設定によってはキャラクタの使用言語の違いを反映して自動的に会話の一部が隠されるようにスクランブルが掛かるようにできます。これはキーワードやランダムなどで指定することができます

キャラクタMUCではルールとシナリオの設定によって会話量や会話時間をゲーム内の時間経過に反映させることができます。これによって時間的に緊迫した状況を演出することができるようになります。

XGMTKでは後述するようにゲーム内の情報交換を促進するため、GMやプレイヤーが各種の情報にブックマークをつけて管理することができます。キャラクタMUCではこのようなブックマークのURLを気軽に交換できる「かくかくしかじか」機能を提供します。例えば二手に分かれた冒険者パーティの一方が何か隠されていた手がかりを発見した場合、それをブックマークすることができ、そのブックマークのURLを合流後に

会話に「貼り付けて」送ることにより手がかりを共有できます。また例えば GM が状況説明を前もって準備した画像や音声、テキスト他のデータで用意してあればそれを状況説明の文に「貼り付けて」送ることもできます。

その他、XMPP がサポートする感情表現の送信といった機能を生かしてリッチなチャットをサポートする予定です。またキャラクタ MUC のログを取ることでリプレイ記事を書きやすくすることができるでしょう。

以上のようにプレイヤーMUC はキャラクタ相互間、キャラクタと GM 間の対話を支援する中核的な機能となります。この節で述べた各機能をユーザ (GM とプレイヤー) に対して実際にどのような形で表現し、操作できるようにするかはクライアント・ソフトウェアの作りと設定に依存しますが、少なくともシステムはこれらの情報を適切に扱います。

ルール駆動対話と行動結果通知

TRPG にはプレイヤーや GM 等「中の人同士の会話」(プレイヤーMUC で行われる) だけでなくキャラクタ間の会話や GM による情景描写 (キャラクタ MUC で行われる) もないカテゴリの対話があります。それは GM とキャラクタの間でゲーム上の行動の表明や行動の成否判定といったようなゲームのルールに従うやりとりです。

通常行われるような、実際に集まって顔を合わせて行う TRPG セッションでこれは通常のインフォーマルな会話に紛れた形で行われています。しかし遠隔セッションでは通常より伝達できる情報が少ないのでこれを通常の会話のような冗長な形式で行うと煩雑なものになってしまうでしょう。

しかし行動の表明についてよく観察してみると TRPG でキャラクタが可能な行動の多くはルールに規定があるため、行動の表明に必要な情報は入力フォームに設定をしてもらうような形で表現できます。従って GM からルールに従ったフォームを送りつけ、その中の必要項目をプレイヤーに埋めて送り返してもらい、必要であればさらに別のフォームを送って追加情報を求めるような形式の通信をサポートすれば TRPG のセッションをスムーズに進めることができるでしょう。XGMTK ではキャラクタの行動の表明に利用されるこのような形式の対話をルール駆動対話と名付けます¹⁴。

またもう一つの行動の成否判定は XGMTK では行動結果の通知と呼びます。行動結果の通知は GM による判定・決定の通知となります。行動結果通知には二つの側面があり、一つはプレイヤーへの説明、もうひとつはデータの更新です。この 2 つの側面に対応し、行動結果の通知は 2 つの経路で行われます。一つはプレイヤーに説明するための情報を通知する経路であり、もう一つはプレイヤー側に置かれたデータを更新するための経路です。前者は状況説明情報としてとしてキャラクタ MUC で送られます。後者のようにデータの書き換えを必要とする通知はソフトウェアによって自動的に処理されます。詳細は「遠隔セッション支援機能」の「回線障害・チート対策とデータ更新」(p.17) で合わせて説明します。

ルール駆動対話及び行動結果の通知はコンピュータ RPG、特に MMORPG や MORPG における行動選択方式

¹⁴ ルール駆動対話は XMPP 的には XEP-0050: Ad-Hoc Commands を通じて行われることになるでしょう。

とデータの更新方式に似たメカニズムであると言えます。後述の自動化支援機能ではこのフォームのやり取りをルールに沿って自動化することでGMの作業量を削減したり、キャラクターの応答や行動を自動化したりします¹⁵。これらルールに則って半自動、または自動で行われるキャラクターとGM間の対話は遠隔セッションという必要から生まれたものですが、遠隔セッションだけでなく、顔を合わせて行うセッションでもスムーズなゲームの進行に役に立つであろうと考えられます。

ダイス

TRPGでは行動の成否を確率的に判定する機会が良くあり、しばしばその解決にはサイコロ（ダイス）が用いられます。ダイスによる判定はプレイヤーに成否を知らせず結果だけを通知する場合（例えばある音が聞こえたかどうかや、移動中に的に遭遇するかどうかなど）と、プレイヤーが自身で運試しする事例（例えば攻撃が命中するかどうかなど）に分かれます。XGMTKではチート防止のため、プレイヤー自身で運試しする場合はルール駆動対話によるプレイヤーのアクション（フォームのダイス・ロール・ボタンを押すなど）のタイミングに基づきつつ、実際のダイス判定は全てGMサイドで行う方式を基本とします。GMはこの時シナリオ難易度のバランス調整の必要を認めればサイの目を任意に変更することもできます。

XGMTKでは各種のダイスを表現する乱数発生器を準備し、ダイスの判定もソフトウェアで行うことができます。ただしプレイヤーとGM間に十分な信頼関係がある場合や、遠隔ではなく参加者が顔を合わせてセッションする場合などでは、物理的なサイコロを振る楽しみのためにプレイヤーが自分で物理的なサイコロを振ってその値を報告するように、GMの権限でセッションを設定することができます。

2-2 データ共有支援機能

データ共有支援はコンピュータRPGの特徴であるリッチなマルチメディアデータとそれらに対するGUIを提供します。しかしそれだけではなく、従来各TRPGでバラバラであったルール、シナリオ、データの表現の標準化をも企図しています。

XMLの採用

XGMTKではデータの共有の枠組みとしてXMLを採用します。XML¹⁶はeXtensible Markup Languageの略で厳密に形式を定義しながらも拡張可能であるような構造的なテキスト・データ形式を表現する言語（マークアップ言語）を定めた規格です。

XMLはHTMLなどと同じく基本的にテキストですが、これにタグと呼ばれる定まった形式を持ち、タグ（tag）と呼ばれる目印となるテキスト（例えばHTMLであれば<a>、やといったようなもの）を

¹⁵ それを極限まで推し進めGMの行動を全てソフトウェアによって自動化してしまえば、XGMTKをMORPGシステムとして利用できるということになります。

¹⁶ 公式サイト（英文）：<http://www.w3.org/TR/xml11/>

追加する（マークアップ）ことでテキスト・データを要素（element）という単位に分割して階層構造を持たせることができます。

HTML は Web ページを表現することに特化したマークアップ言語ですが、XML は汎用であり、ユーザが目的に応じて必要なタグの種類と構造を定義することができます（そのようにしてユーザが定義したタグとその使い方の決まり一式によって定義される文書形式を XML 語彙と呼びます。）。XML は名前空間と言う仕組みで複数の XML 語彙を組み合わせることもできます。XML は今日では様々なデータの表現に用いられています。特に前述の XMPP では、通信の開始から終了までを特定の XML 語彙に従う「文書」と見なすことにより、厳密かつ拡張可能なプロトコルを定義するために利用されています。

XGMTK では TRPG を構成する様々な要素、例えばキャラクタ、アイテム、エリア、イベントからシナリオ、ルールに至るまでの全てを XML 要素で表現するための XML 語彙を定めます。そして各 TRPG 固有のルール、世界設定に関する部分を名前空間の仕組みを使って拡張可能なものとします。

XML を解析してデータ構造に変換することを助ける XML パーサというライブラリも充実しており、XMPP でも XGMTK でもそれらを活用して XML 表現をプログラムで扱いやすいデータ構造に変換して処理します。

データ形式と GUI の分離

XGMTK では TRPG で利用されるあらゆるルール、シナリオ、データ、マルチメディア素材を全て XML 語彙として定義します。これら XGMTK のための XML 語彙はデータの形式と意味を表現しますが、ユーザ（プレイヤーや GM）に対し、それをどのように表示し、操作させるかは定義しません。例えば 2 次元のマップのデータがあった時、これをどのように表示して、移動の仕方をどう指定させるかはクライアント・ソフトウェア（ビューワーと呼びます）に任せられます。ビューワーはシンプルな平面図で表現してもいいし、適当に 3 次元化して表示してもいいわけです。

このように決めるのはデータ自身よりその表現や操作方法の方が技術革新の影響を受けやすいという経験に基づいています。またこうすることでユーザが使いやすい表示方法や GUI を選ぶことができるようになります。HTML とウェブ・ブラウザの関係などがそのような例として挙げられるでしょう。

回線障害・チート対策とデータ更新

遠隔セッションでは不意の回線切断時のデータの整合性に配慮する必要があります。また、プレイヤーのチート対策にも配慮する必要があります。

チートはゲーム性を薄めプレイの緊張感をそぐため TRPG においてもあまり歓迎できるものではありません。TRPG は基本的にプレイヤー間では協力ゲームになることが多いのでチートはそれほど起こらないのではなかろうか…と感覚的には思うのですが、残念ながら、失敗を恐れるあまりか、見栄を張りたいのかどうか TRPG でもたま〜にチートする人がいます。例えば、妙なサイコロの投げ方で出目を操作しようとしたり、キャラクタ・データの更新でズルをしたり、GM の手元の資料を盗み見たりといったような行為をするプレイヤーを筆者は見かけたことがあります。非常に残念なことですが、GM や他のプレイヤーが目の前で見ていてさ

えそうなので、遠隔セッションではチートへの誘惑は高まる可能性があると考えられます。

よって回線切断地のデータ不整合、チート対策に配慮するため、データ共有に際してルールやシナリオ中、書き換え可能で、ゲームの進行に影響を及ぼすようなデータ、例えばキャラクタ、アイテム、マップ、シナリオについては以下のように管理します¹⁷：

- 1) GM 側にマスター・データを置く
- 2) 表示や GUI を提供するのに必要なコピーを各プレイヤーに配布
- 3) 全てのデータ書き換えはマスター・データをまず書き換え、その後プレイヤー側に書き換えを通知して更新
- 4) 書き換え完了通知でプレイヤーは対象データの SHA-1 ハッシュ値を報告
- 5) また回線切断など不整合が危惧される場合にはプレイヤーは不整合が怪しまれるデータ（プレイヤー側に置かれている全てのデータかもしれない）の SHA-1 ハッシュ値を報告
- 6) ハッシュ値の不一致があった場合はプレイヤー側のコピーを現在のマスター・データに置き換える

汎用性

XGMTK は特定の TRPG のルール、システムに依存しない環境、ツールを目指しています。従ってデータ表現も特定のルールに依存しない標準的なものにする必要があります。これを実現するために XGMTK は多くの TRPG のシステムの中で中立的なデータ表現が可能なものは極力中立的に表現します。そしてそれ以外の部分についてはルール自身をデータとして記述し一種のスクリプト/プログラムとして取り込むことで各種 TRPG のルール、シナリオ、データを表現します。

幸い TRPG のルールシステムを観察すると単位系、時間、空間表現（フィールドやエリア）の多くはゲームのルールから中立であるか、中立的な表現からの変換が可能であるとわかります。これは TRPG の世界観が基本的に人間の日常的なスケールで構築されていること、およびシナリオや素材の作成の手間やプレイしやすさの観点から、あまりルールでガチガチに縛らずマスターの裁量に任されている部分が多いことに拠っていると考えられます。これは多くのコンピュータ RPG では GM をプログラムで代用する必然性からそのあたりもガチガチに決まっていることと好対照だと言えます。

一方、キャラクタの能力、スキル（ファンタジーRPG においては魔法も含む）とキャラクタが可能な行動の選択肢、行動で利用されるアイテム（武器や防具、日用品、宝物や貨幣）、は TRPG のルールではそれぞれ事細かに規定されています。これは TRPG のルールがキャラクタの生活と戦いをどうパターン化／モデル化するかに重点を置いて設計されており、各ルールがその点で差別化されていることの表れだと考えられます。

¹⁷ XMPP 的にはこの手順は XEP-0009: Jabber-RPC および XEP-0163: Personal Eventing Protocol か XEP-0060: Publish-Subscribe を通じて行われることになるでしょう。

以下の節ではこれらのデータをどういう方針で取り扱うかについて述べます。

単位系

ゲーム内の各種単位系としては各 TRPG のゲームシステムに中立な MKS 単位系を採用し、倍精度浮動小数点数 (Java 言語でなら double 型整数) で管理します。各 TRPG での表示に合わせるため、各 TRPG のルール記述内にはこれらの単位系からの変換テーブルを含めます。ただし、ゲーム内時間の管理だけは後述のイベント管理との関係で例外となります。

またアイテムの中で計量可能なもの、例えば各種の液、粒、粉末や、長さのあるロープといったものについてはさらにそれらのアイテムの単位量から重さや強度、体積、価値といった別の単位量に変換する情報をアイテムのデータに含める必要があります。

ゲーム内時間

ゲーム内の時間の管理はゲーム内のイベントの管理と密接に結びついています。イベントについての詳細は後述しますが、ここでは時間と関係ある部分について述べます。ゲーム内のイベントの順序は因果関係を保存するため正確である必要があります、誤差を許容しません。このためその他の単位とは異なり、秒単位の整数 (Java 言語でなら long 型整数) で管理します。表示のためにゲーム内の単位への変換規則は他の単位と同じくルール記述内に変換テーブルを用意します。ゲーム内の単位としては世界観に関わるもののほかにルールで利用される時間単位であるターンやラウンドと言ったものも含まれます。

また日付、曜日、年号等を表現する暦は各ルール、世界設定あるいはシナリオに依存するため各 TRPG のルール記述内やシナリオ内に変換テーブルや変換式を用意する必要があります。

XGMTK において一人の GM が管理する一つのセッションは 1 つのイベント・マネージャを持ちます。イベント・マネージャは開始時刻と経過秒数によってゲーム内における現在時刻を管理します。セッション開始時刻はゲーム内時間での開始時刻表し、ルール (あるいは世界設定やシナリオ) によってきまる暦上の時刻の形式で保持されます。経過秒数はその開始時刻から何秒が経過したかを表します。経過秒数はそのセッションに参加するキャラクターの各種の行動に応じて加算され、それによってゲーム内時間が経過していきます。

イベント

XGMTK ではゲームの進行はイベント連鎖として表現され、管理されます。例えば先に述べたルール駆動対話と行動結果通知 (p.15) 自身も一連のイベントとして管理されます。また後述するようにルールやシナリオも再利用可能なようにパターン化されたイベントの連鎖として表現されます。

イベントはセッションに一つあるイベント・マネージャに登録され、ある条件を満たすと発生します。イベントが発生するとイベントに関連付けられたスクリプト (イベント・スクリプト) が実行されます。スクリプトについては自動化機能で後述します。スクリプトでは以下のような処理が可能です：

- 様々な計算処理

- キャラクタ MUC へのメッセージ発信（行動結果通知・メッセージ発信）
- GM やプレイヤーへの問い合わせ処理（ルール駆動対話）
- データ更新処理（行動結果通知・データ更新）
- 新たなイベントをイベント・マネージャへ登録

イベントには以下のような種類があります：

- 時刻を条件として発生するイベント（**時刻イベント**）
- 設定された条件が満たされると発生するイベント（**条件イベント**）

時刻イベントは指定した時刻になると発生するイベントです。時刻イベントはマネージャ内で時刻順に並べたリストで管理されます。イベント・マネージャは前述のように時刻も管理しており、イベントの時刻順リストで現在時刻に直近のイベントを選び、そのイベントの発生時刻まで現在時刻を進めて、イベントを発生させ、イベント・スクリプト終了後に廃棄します。つまり時刻イベントはイベント発生によって消費されます。また過去の時刻についての時刻イベントを登録するとエラーになります。このためイベント・マネージャは絶えず現在の時刻以降の時刻イベントだけを保持することになります。

同時刻のイベントの発生順序は優先順位を示す整数によって決まります。優先順位は一般に 0 以上の整数ですが-1 は特別で優先順位最低を意味し、**LOWEST_PRIORITY** という名前を持ちます。**LOWEST_PRIORITY** は一般に行動結果通知としてイベントの結果をデータに反映させる際、同時に発生した他のイベントに副作用を発生させないようにイベントデータを更新する目的で使われます。時刻イベントがイベント・スクリプト内で現在時刻にイベントを登録する際は自身の優先順位より低い優先順位のイベントのみを登録することができます。

優先順位の値は対象となる **TRPG** のゲームシステムによっては時間としては区分できない同一時の事象であるが、その結果判定に優先順位が定まっている場合を表現するために利用します。例えば 1 ラウンドの攻撃を敏捷性(DEX)の数値の順に行う **D&D** の優先順位付けや **RuneQuest** で 1 ラウンド内の攻撃順序を規定する **ストライク・ランク (SR)** 値がこれに該当します。優先順位を示す数値は各ルールに従ってイベント登録時に与えられます。

時刻イベントは特定の時刻の一つにのみ関連付けられています。定期的発生するイベントのためにはイベント・スクリプト内で次に発生すべきイベントを登録します。遅延して発生するイベントがある場合も同様です。例えば 24 時間で **HP**（体力を表す数値）が満タンに回復することを表現するには減っている分の体力値を 24 で割った分だけ **HP** が回復するようなイベントをイベント・スクリプト内で 1 時間後に登録することを繰り返すような方法が考えられます。

条件イベントはイベント・マネージャが現在時刻のイベントをすべて発生させ終え、時刻を更新した直後に条件をチェックし、条件が満たされていれば発生します。条件の判定には実行後真偽値が返される式であるよ

うなスクリプトが実行され、その結果が真であった場合に発生します。条件イベントは時刻イベントと異なり発生後にイベント・スクリプトで明示的に消された場合だけイベント・マネージャから廃棄されます。

時刻と条件の組み合わせなどより複雑なイベントは上記 2 種類のイベントをイベント・スクリプトで組み合わせて実現します。例えばある時刻に条件判定をしたい場合は、時刻イベントのイベント・スクリプト内で条件を判定し、条件が満たされていない場合は何もしないで終了することで実現できます。

空間

XGMTK ではあらゆる空間は**エリア**として表現されます。エリアは宇宙、惑星系、世界全体から、大陸、海洋、島、地域、特定の場所、街、建物、部屋など様々なスケールの空間を表現します。変わった用途としてはエリア用 GUI を流用して宝箱の中、絵画や図といった小さな空間をもエリアとして表現し、キャラクタにその中に隠されたアイテムや手掛かりを探させるといった応用も想定されています。

一般に XGMTK のエリアはそこに置かれた様々な対象のおかれた位置を表現します。そのためエリアにはエリア内の位置を示す座標の表現があります。エリアはキャラクタ (PC、NPC)、サブエリア (後述)、そして条件イベントといったエリア・オブジェクトを配置することができます。この条件イベントを通じて手がかりとなる情報やアイテム、他のエリアへの入り口／出口、罠といったものをキャラクタに遭遇・発見・利用させることができます。またエリアに対する GUI はキャラクタの移動を制御する手段 (例えば可能な目的地や経路の選択肢を表示して選ばせるなど) を提供します。

通常 ¹⁸エリアは別のエリアをサブエリアとして持つことができ、それによる階層構造が組み立てられます。一つのセッションには階層構造のトップレベルのエリアがただ一つ保持されます。トップレベルのエリアの名前は必ず”Area”となります。エリアはそれ自身の名前を保持していて、ファイル・システムのディレクトリのように”/”で区切って並べることで階層内の特定のエリアを指すことができます。これをエリアパス (Area path) と呼びます。エリアパスと上述したエリア内での座標の組み合わせでキャラクタ等の位置を一意に指定することができます。各エリアは階層構造で親にあたるエリア内での位置を自身の基準位置として保持します。またエリアの内部データの単位長さが何mになるかを示すスケールの情報を持っています。

エリアはエリア内のキャラクタに対して環境を提供します。環境としては明るさ、天候 (気温、風、降水、気圧など)、植生、足元の状態などが考えられます。エリアはエリア内の位置に応じて環境を変えることができます。また時間イベントと組み合わせて時間によって環境を変化させることもできます。エリアは階層構造で親にあたるエリアから環境を継承します。エリア内で環境の情報を定義することで継承した環境を選択的に上書き修正できます。

エリアではエリア内の座標やエリアパスで表現される 2 点の距離と方角を算出することができます。この機能は飛び道具の射程距離判定や、キャラクタ MUC (p.14) で会話や音などの情報がどの程度届くかを判定するためにキャラクタ間の距離を測るために利用されます。

¹⁸ つまり GUI を流用するような応用例以外で、キャラクタの位置を管理する場合。

エリア・タイプ

エリアは表現や GUI のリッチさとオーサリングの手間とのトレードオフに配慮して複数のタイプの併存を想定しています。複数のタイプのエリアは様々なスケールを持ち、エリアの階層構造のどこにでも入れることができ、全体としてセッションに複数のタイプ、スケールのエリアが混在することになります。

このようなタイプ、スケール混在の階層高構造を採用したのは以下のような考察に基づきます。TRPG には時として長い旅の過程が含まれることがあり、その全過程を全て同じ情報密度で表現することは必ずしも必要ではないはずです。幾つかのコンピュータ RPG では大陸全土を 2D や 3D マップで一様に表現して端から端まで歩いて移動できるものがありますが、これはスケールのかなり無理がある言わざるを得ません。

例えば人気のある MMORPG である Final Fantasy XI (FF11) では通常の 3D フィールドに比べればかなり広大な 3D マップを連結して大陸全土を表現し、リアルな時間経過も設定されていますが、残念ながら「大陸」の端から端までをゲーム内時間で測って一昼夜もあれば人間の脚で踏破できてしまうようなことが発生しています。とはいえ、これが実際にゲーム内時間で何カ月もかかるようであれば当然著しくゲームの遊びやすさが低下することでしょう。

一様な密度でエリアを構成することには作成コストの問題もあります。FF11 のような広大な 3D マップを大陸一つ分作ることは大企業で人手とお金をかけなければできないものではありません。よって現実的な解はシナリオ上重要な部分には手をかけてスケールが小さく密度の濃いエリアとして作成し、その間を結ぶ部分には大スケールで情報密度が薄いエリアを使うことです。(実際は FF11 でも通常のフィールドと街やダンジョンでは微妙にスケールを変えているようではありますが…)

現在 XGMTK で開発案のあるエリア・タイプは以下の通りです：

- Environment
- Linked
- Stacked
- D2
- SectionedD2
- DividedD2
- LayeredD2
- D3

Environment は広さを持たない特殊なエリアです。座標は 1 点だけで、そこに置かれた各エリア・オブジェクトはすべて互いに等距離でスケールの示す単位距離に等しいと仮定されます。単位距離を 0 とすると全てが同一点と言うことになります。方向は意味を持ちません。このタイプのエリアはエリア階層の下位に対して

デフォルトの環境を提供することが主な目的となります。想定されるグラフィカルな表示法はありません。

Linked はリンクされた各点によるグラフで表現されるエリアです。各点においてリンクには方向や名前を結び付けることができます。各リンクには距離を結び付けることができます（その際幾何学的な矛盾はチェックされません）。距離が指定されていない場合はスケールの示す単位距離に等しいと仮定されます。直接リンクされていない各点間の距離は最短経路から計算されます。座標は各点の名前です。同一点におかれたエリア・オブジェクトは全て同一地点にあると見なされます。

このタイプのエリアは往年のテキスト・アドベンチャーで使われていたような離散的なマップを表現することが主な目的となります。想定されるグラフィカルな表示法はありませんが、エリア・オブジェクトとして条件イベントを設置し、その条件オブジェクトのイベント・スクリプト内でテキストや画像の表示を設定することでリンクされている各点にたどりつくたびにその点に結び付けられたテキストや画像を表示させることができます。見かけは極めて貧弱ですがエリアの作成は非常に簡単ですので、シナリオの試作やシナリオ上の重要な地点間を結ぶ糊のように使うことができますでしょう。

Stacked は **Linked** に似ていますが、構造はさらに簡単です。**Stacked** では単純にエリア・オブジェクトを同一地点に積み重ねます。座標は整数一つで、それにより垂直方向の高さ／深さが決まります。置かれたエリア・オブジェクトの水平方向の位置は全て同じになります。垂直方向の距離は座標値が 1 違うとスケールの示す単位距離だけ離れていると仮定されます。このタイプのエリアは建造物や地下迷宮の各フロアを表現するエリアを積み重ねて垂直方向に伸ばすために利用するのが基本的な用途です。

D2 は HTML で言うところのクリックابل・マップのようなもので、1 枚の画像上に点を指定してエリア・オブジェクトを配置できます。画像を一枚描いて、そこにエリア・オブジェクトを配置するだけという非常にシンプルな方法でエリアを作成できます。エリア・オブジェクトの水平方向の位置と方角は管理されます。環境はエリア内の全域で同じになります。

SectionedD2 は升目（スクエアやヘックス）が規則正しく並んだものとして表現されるエリアです。点が規則正しく配置されることと、各升目に表示用のアイコンと基準点からの高度差を設定できること以外は **Linked** と同等です。座標は升目を表す 2 つの整数の組です（スクエアでは自明ですし、ヘックスの場合は縦に半升ずつずれたスクエアとして座標をつけることができます）。スケールで示される単位距離は隣接する 2 つの升目の重心間の水平方向の距離となります。拡張として各点に結びつけた **Environment** を解釈することで高度差と合わせてストラテジー系のゲームでおなじみの「地形効果」のようなものを表現することもできます。

古典的に多くのストラテジー系のテーブルゲームで利用されてきていてユーザによく親しまれており、ユーザにとっては距離や方角が目測しやすいという特徴があります。ゲーム進行中にスクリプト言語により動的な生成をすることも簡単なため、戦略的性が重要な戦闘の場合に、一時的なエリアを生成して用いることもできるでしょう。

DividedD2 は境界線で区切られた複数の地点からなるエリアです。内部的には **Linked** と同じですが 2 次元

画像が結び付けられており、D2 のように点を指定すると対応する地点を選べます。各種行政区画や国境線等といった不規則な境界線で区切られた地図で表現されるようなエリアを作成するのに向いています。

LayeredD2 はベースとなる 1 枚の画像に各種の属性を同サイズ複数のグレースケール画像のマップとして重ね合わせて作成される 2 次元のマップとして表現されるエリアです。植生マップや高度マップ、気候マップ、街道マップ、建築物マップ、移動コストマップを始め様々な情報を重ね合わせることができます。これらのマップの種類は拡張可能です。エリア・オブジェクトの設置については 2D マップと同じです。

情報の表示はビューワーに任せられ、ビューワーは理解できない拡張マップの情報は無視します。ビューワーはエリアを真上から見た 2 次元画像として表示することもできますし、重ねあわされた高度マップや植生マップといったマップの情報をを用いて 3D の地形表示（CG 技術を駆使すれば俯瞰や、ファースト・パーソン・シューティング風の表示も可能でしょう）をすることもできます。

以上から **LayeredD2** は一般的なオープン・フィールドの表現に向けたタイプのエリアと言えるでしょう。天井の高さや壁といった適当なマップ拡張を用意してビューワーの実装を工夫すれば建築物の内部や地下迷宮など壁や天井のあるエリアでも表現できるかもしれません。

エリアの作成は比較的簡単ですが、マップの枚数が増えるとそれに応じて手間がかかるようになるでしょう。

D3 はいわゆる 3D グラフィックス表現されるようなタイプのエリアを想定しています。VRML など既存の規格の採用も考えられますが、現時点で詳細は未検討です。恐らくビューワーの実装も移動のための GUI の実装も難しく、特にエリア作成の手間が相当であろうと推測されます。このため将来の課題とし、当面の実現は見送ることになるでしょう。

キャラクタやアイテム

キャラクタやアイテムが保持するデータは各 TRPG のルールによってかなり内容が異なります。具体的な要素の構成は各ルールの記述に従うことになるでしょう。ただ前述のイベント連鎖のパターンで各種の行動が類型化されており、キャラクタやアイテムはその行動の選択や結果の判定のために受動的にデータを参照され、書き換えられる対象となると想定されます。スキルや魔法などはイベント連鎖のパターンと関連付けられることになるでしょう。

秘密の管理

TRPG のだいたい味の重要なものに秘密の探索があります。秘密としては隠された財宝や情報、隠された通路や部屋といった存在が挙げられるでしょう。秘密は条件イベントとして記述され、エリアやキャラクタ、アイテムに付加されなんらかの条件を満たすことでその条件を満たしたキャラクタ（場合によってはそこに居合わせた他のキャラクタ）に存在が開示されます。

特殊なタイプの秘密としてはエリアの未探索部分の情報を隠すといったようなものもあります。これをどのように実現するかは各エリアのタイプとビューワーの実装に依存します。

2-3 自動化支援機能

XGMTK は基本的に人間の GM がゲーム進行とレフェリングを司る伝統的な TRPG の支援を目的にしています。しかしコンピュータ・ソフトウェアによる自動化を全く排するというものではありません。ソフトウェアによるゲーム進行の自動化が役立つ局面は以下のように幾つか想定できます：

- ルールのスクリプト化
- GM 応答のスクリプト化
- プレイヤー応答のスクリプト化
- キャラクタ・メイキング
- シナリオのスクリプト化

ルールのスクリプト化とは以下のようなものです。TRPG のルール中には、生活シミュレーションや戦闘シミュレーション的観点でのリアリティのために導入されたものの手順が煩雑であるようなものがあります。この手順が決まり切ったものであれば自動化が可能で、かつスムーズなゲーム進行のために望ましいと考えられます。「遠隔セッション支援機能」の「ルール駆動対話と行動結果通知」(p.15) で考察したように、TRPG のシステムにおける行動表明や戦闘の解決手順の多くは、決まった手順で決まった形式のフォームを提示して値を設定してもらうことを繰り返して進行可能です。この決まった手順で決まったフォームを提示し結果を受け取って判定を行うというのは、ソフトウェアによる自動化になじむ部分となります。XGMTK では「データ共有支援機能」の「イベント」(p.19) で提案したイベントの仕組みを利用することでルールに含まれる手順をスクリプト化します。

GM 応答のスクリプト化は、GM が管理するキャラクタである NPC の内、シナリオ上あまり重要でない NPC の行動（アイテム販売をする店主など）や雑魚敵などある程度自動化しても支障のない存在について応答を自動化します。具体的には GM を介して送られてきたフォームに状況に応じた値を設定し、プレイヤー応答スクリプトキャラクタ MUC に定型のメッセージを表示するといったようなスクリプトになります。

プレイヤー応答のスクリプト化はメカニズムとしては GM 応答のスクリプト化と同じです。ちょっとした離席やあるいは欠席といった場合にユーザが自動応答スクリプトを記述できれば、セッションに参加するプレイヤーの一部が一時的な離席した場合や欠席をした場合にもセッションを成立させることができるでしょう。欠席した場合にパーティメンバーに加え探索や戦闘にも参加させるのか、探索に赴くパーティとは別行動で日常的な労働や訓練をさせるのかはプレイヤーが作成するスクリプトの内容によるでしょう。特殊なスクリプトとしては、毎回の定型的な入力を避けるため前回の応答を繰り返す、他キャラクタの行動を模倣し追隨する、一時的にキャラクタの操作を他プレイヤーや GM に委ねるため、GM や他プレイヤーへ応答を中継するというようなスクリプトの可能性も考えられるでしょう。

キャラクタ・メイキングのスクリプト化はメカニズム的にはルールのスクリプト化の特殊な場合で、キャラ

クタ作成を自動化します。フィールドやダンジョンでのランダムな遭遇のためにランダムにキャラクタを作成するというような応用も考えられます。

シナリオのスクリプト化はルールのスクリプト化の特殊な場合と考えられます。TRPG ではルールがカバーしていない事象の処理についてシナリオでローカルルールを設定することが良くあります。このためシナリオはローカルにルールを定義する仕組みを備えている必要があると考えられます。また複数のシナリオでキャンペーンを構成することや、世界設定を共有した複数のキャンペーンやシナリオを作成することもよくあります。このため事実上シナリオとルールに明確な境目はなく、基本的なルールから世界設定、キャンペーン、個々のシナリオというような階層構造をなすものとするのが適当でしょう。そしてこの階層構造では基本的なルールに近いほど再利用可能なパターンが多く、シナリオに近いほどパターンを引用して具体的な値を設定したパターンのインスタンスが多くなると考えられます。

スクリプト言語の実装

スクリプトの構文規則については未検討です。既存のスクリプト言語（JavaScript など）の流用や XML の要素を使った構文（XML のスタイルシートの規格である XSLT に見られるような）、あるいは両者の融合様々な可能性を検討しています。2011 年末までに一通りの検討をしてみたいと考えています。

とはいえ参照や操作の対象が XML であることから XPointer や XPath といった XML の要素を指定して値を取り出すための構文規則を利用することになる可能性が高いでしょう。

またスクリプト言語でルールやシナリオを記述するとゲーム進行のスムーズさがそのスクリプト言語の解釈実行の速度に左右される可能性があります。もしスクリプト言語の実装後、そのような問題があれば、スクリプトを Java のバイトコードへコンパイルするような仕組みについて検討する必要があるかもしれないと考えています。

2-4 オーサリング支援機能

TRPG の楽しみはプレイすることだけではありません、GM としてシナリオを作成しセッションを主催、進行することも大きな楽しみです。よってこれを支援するためシナリオの作成を助けるオーサリング・ツールは必須でしょう。オーサリング・ツールが扱うべき対象は大きく分けて以下のような形式に分かれるでしょう：

- XML ファイル
- エリアやキャラクタ・イメージに利用する画像素材
- スクリプト言語

現時点ではまだデータ共有のための XML 語彙の詳細が未定なのでオーサリング・ツールに求められる機能も未定です。またエリアなどのビューアの機能を開発ツールが模倣して出来上がりを確認できると望ましいので、各種ビューアの仕様がひとまず固まるまではあまり詳細を決められないという事情もあります。

とはいえ将来的な発展の方向性に関する一つのヒントとして **Rune Quest** のルーンクエスト・シティーズ(原題: **Rune Quest Cities**) というサブプリメントは注目に値するでしょう。ルーンクエスト・シティーズは文化人類学的な研究を反映した汎用の架空の都市デザイン手法のガイドになっていて様々な規模の都市のデザインをそれらしく作り上げる手順が解説されています。このようにあるエリアを様々な法則に従うそれらしい姿に仕上げるような機能を持つとオーサリング・ツールの使い勝手が向上するのではないかと期待されます。例としては浸食効果をエミュレートして自然な地形を作るとか、歴史上の様々な建築スタイルを表現する部品のセットを用意するといったようなことが挙げられるでしょう。

3 開発目標と方針、計画

以上、XGMTK の企画案は開発言語に依存しませんし、各機能が異なる言語で書かれた異なるソフトウェアで実現されていても問題はありません。遠隔セッションで複数の異なる XMPP クライアントが混在しても XGMTK の定める仕様さえ満足していれば問題はありません。実際、オーサリング支援機能と遠隔セッション支援機能は異なるソフトウェア/ツールとして実現され、データ共有支援機能で定められる XML 表現がそれを仲立ちすることになるでしょう。

とはいえ筆者としては当面、開発言語としてはJava (Ver.6)¹⁹、XMPPライブラリとしてはSmack²⁰、XMPPサーバとしてはOpenfire²¹、XMLパーサについてはクライアント環境ではXML Pull Parsing²²、オーサリング支援ツールではXOM²³を利用する予定です。またオーサリング・ツールはEclipse IDE²⁴の上にプラグインとして実装することになるかもしれません。

プロジェクトの Web サイトは <http://www.xgmtk.org/> です。Java のパッケージ名は org.xgmtk を利用します。

開発は大まかに以下の順に進める予定です：

- 1) XMPP 上での基本的な通信機能をサポートするフレームワーク
- 2) 1)を利用した対話環境を XMPP クライアント・プロトタイプの開発
- 3) 自動化支援のためのスクリプト言語の開発
- 4) データ共有支援機能のための XML 語彙の詳細設計
- 5) データ共有支援機能のための GUI 開発

¹⁹ <http://java.sun.com/javase/ja/6/download.html>

²⁰ <http://www.igniterealtime.org/projects/smack/index.jsp>

²¹ <http://www.igniterealtime.org/projects/openfire/index.jsp>

²² <http://www.xmlpull.org/>

²³ <http://www.xom.nu/>

²⁴ <http://www.eclipse.org/> 日本語 Wiki サイトは <http://www.eclipsewiki.net/eclipse/>

6) オーサリング支援ツールの開発

この中で 10)についてはサンプルとして **RuneQuest** のルール記述とサンプルシナリオを作成する予定
希望的な予定としては、今年末までに 2)、来夏までに 4)までが完了していきたいと考えています。