

スマートスクール・プラットフォーム技術仕様

システム間データ連携インターフェース

技術仕様解説

令和2年3月31日
総務省

< 目 次 >

1. はじめに	1
1.1 本書の目的	1
1.2 本書の構成	2
1.3 本書の前提条件	6
1.4 用語集	7
2. スマートスクール・プラットフォームの全体概要	9
2.1 スマートスクール・プラットフォーム実証事業の概要	9
2.2 スマートスクール・プラットフォームの目的	10
2.3 スマートスクール・プラットフォームの構成	12
2.4 連携対象とするデータについて	13
2.5 システム間データ連携の方法	15
2.6 情報セキュリティの確保について	16
2.7 個人情報保護について	17
3. APIサービスの開発	18
3.1 APIサービスの全体概要	18
3.1.1 スマートスクール・プラットフォーム	18
3.1.2 APIサービスの機能	21
3.1.3 APIサービスのデータモデル	23
3.1.4 共通コード	27
3.1.5 基本項目	31
3.2 児童生徒情報APIサービス	32
3.2.1 仕様概要	32
3.2.2 機能概要	37
3.2.3 静的構造	39
3.2.4 動的振る舞い	44

3.3 教職員情報APIサービス	47
3.3.1 仕様概要	47
3.3.2 機能概要	51
3.3.3 静的構造	52
3.3.4 動的振る舞い	53
3.4 学級情報APIサービス	57
3.4.1 仕様概要	57
3.4.2 機能概要	61
3.4.3 静的構造	62
3.4.4 動的振る舞い	63
3.5 異動情報APIサービス	66
3.5.1 仕様概要	66
3.5.2 機能概要	72
3.5.3 静的構造	74
3.5.4 動的振る舞い	77
3.6 出欠情報APIサービス	83
3.6.1 仕様概要	83
3.6.2 機能概要	87
3.6.3 静的構造	89
3.6.4 動的振る舞い	91
3.7 時間割情報APIサービス	94
3.7.1 仕様概要	94
3.7.2 機能概要	102
3.7.3 静的構造	103
3.7.4 動的振る舞い	104
3.8 日常所見情報APIサービス	107
3.8.1 仕様概要	107
3.8.2 機能概要	113
3.8.3 静的構造	113
3.8.4 動的振る舞い	116
3.9 保健室利用情報APIサービス	119
3.9.1 仕様概要	119
3.9.2 機能概要	122
3.9.3 静的構造	123
3.9.4 動的振る舞い	125
3.10 学習記録情報APIサービス	128
3.10.1 仕様概要	128

3.10.2 機能概要.....	130
3.10.3 静的構造.....	143
3.10.4 動的振る舞い.....	150
3.10.5 ケーススタディ.....	154
4. APIサービスの利用	175
4.1 APIサービスによるシステム間データ連携	175
4.1.1 システム導入時	176
4.1.2 年次処理	179
4.1.3 日次処理	182
4.2 システム間データ連携によるデータ活用事例	194
4.2.1 【活用事例①】 不安を抱える子供の早期発見・支援や児童生徒指導上の早期発見・早期解決.....	197
4.2.2 【活用事例②】 授業を欠席した子供に応じた学習指導.....	200
4.2.3 【活用事例③】 学習理解度に応じた学習指導	202
4.2.4 【活用事例④】 システム管理業務の効率化	204
5. 巻末付録	207
5.1 補足資料	207
5.1.1 全体ユースケース	208
5.1.2 全体データモデル	209
5.1.3 全体データモデル（カテゴリ別）	210
5.1.4 ステートメント全体構成	211
5.1.5 コンテンツメタデータ	212
5.1.6 結果データ	215
5.2 参考資料	217
5.2.1 全体ERモデル（論理モデル）	217
5.2.2 全体ERモデル（物理モデル）	218
5.3 参考文献	219

1. はじめに

1.1 本書の目的

本書は、総務省が教育情報化の一環として推進する「スマートスクール・プラットフォーム」におけるシステム間データ連携インタフェースの技術仕様（以下、「スマートスクール・プラットフォーム技術仕様」）について、

- ① システム設計・開発会社に所属し、校務系システムまたは授業・学習系システムの設計、開発に携わる「システム開発担当者」
- ② 地方自治体（教育委員会）に所属し、校務系システムまたは授業・学習系システムの調達に携わる「システム調達担当者」

を対象に、『スマートスクール・プラットフォーム技術仕様＜技術仕様＞』（以下、「技術仕様」）に記載されている API サービス（以下、「API サービス」）の設計、開発および調達の際に必要な技術的要点を解説したものである。

本書は、対象読者ごとに、以下に示す目的を設定している。

- ① 主にシステム開発担当者を対象として、スマートスクール・プラットフォーム技術仕様に準拠した API サービスを設計、開発するために、オブジェクト指向分析設計の視点から、個々の API サービスの静的構造および API サービスの動的振る舞いを解説する。
- ② 主にシステム調達担当者を対象として、校務系データ、授業・学習系データを調達する上において把握しておくべき API サービスによるシステム間データ連携方法および主なデータ活用事例を解説する。

表 1-1 本書の対象読者

対象読者	説明
システム開発担当者	主にシステム設計・開発会社に所属し、校務系システムまたは授業・学習系システムの設計、開発に携わるシステム設計、開発担当者。
システム調達担当者	地方自治体（教育委員会）に所属し、校務系システムまたは授業・学習系システムの調達に携わる調達担当者。

1.2 本書の構成

本書の構成は以下の通りである。

第1章は、本書の目的、構成、前提条件、用語について記載している。

第2章は、スマートスクール・プラットフォームの全体概要について解説している。

第3章は、API サービスの全体概要について解説したあとに、各 API サービスについて、機能概要、静的構造、動的振る舞いを解説している。

第4章は、API サービスによるシステム間データ連携方法、データ活用事例について解説している。

本書では、スマートスクール・プラットフォーム技術仕様 API サービスについて、①「API サービスによるデータ連携」および②「データ連携によるデータ活用事例」の2つの視点に分けて、①については第3章、②については第4章で解説している。（「図1-1 本書における解説の視点」）

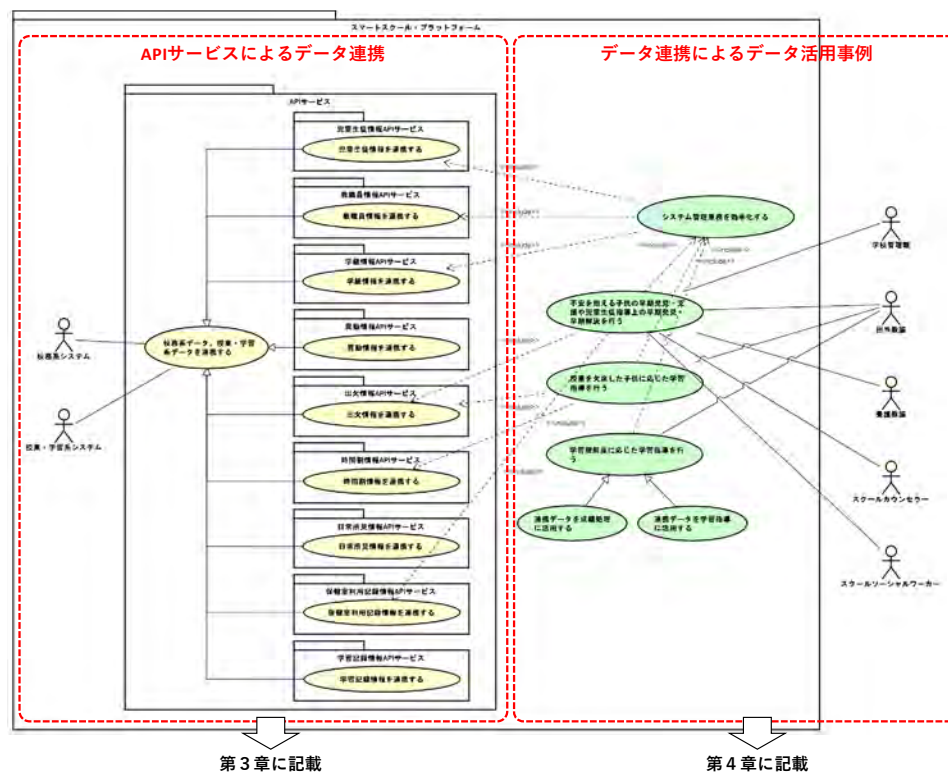


図 1-1 本書における解説の視点

第3章では、オブジェクト指向分析設計の考え方をもとに、それぞれのAPIサービスにおいて、必要となるクラスおよびクラス間の相互作用によりどのようにAPIサービスの機能を実現するのかを主眼に解説している。それぞれのAPIサービスについて、技術仕様に記載されている機能要求仕様を実現するための静的構造および動的振る舞い（論理モデル）を、それぞれクラス図、相互作用図として表現している。第3章では、システム開発担当者がスマートスクール・プラットフォーム技術仕様に準拠したシステムを設計、開発する際に、技術仕様と設計との間をつなげる論理的なモデルを提供することを主眼としている。（図1-2）。

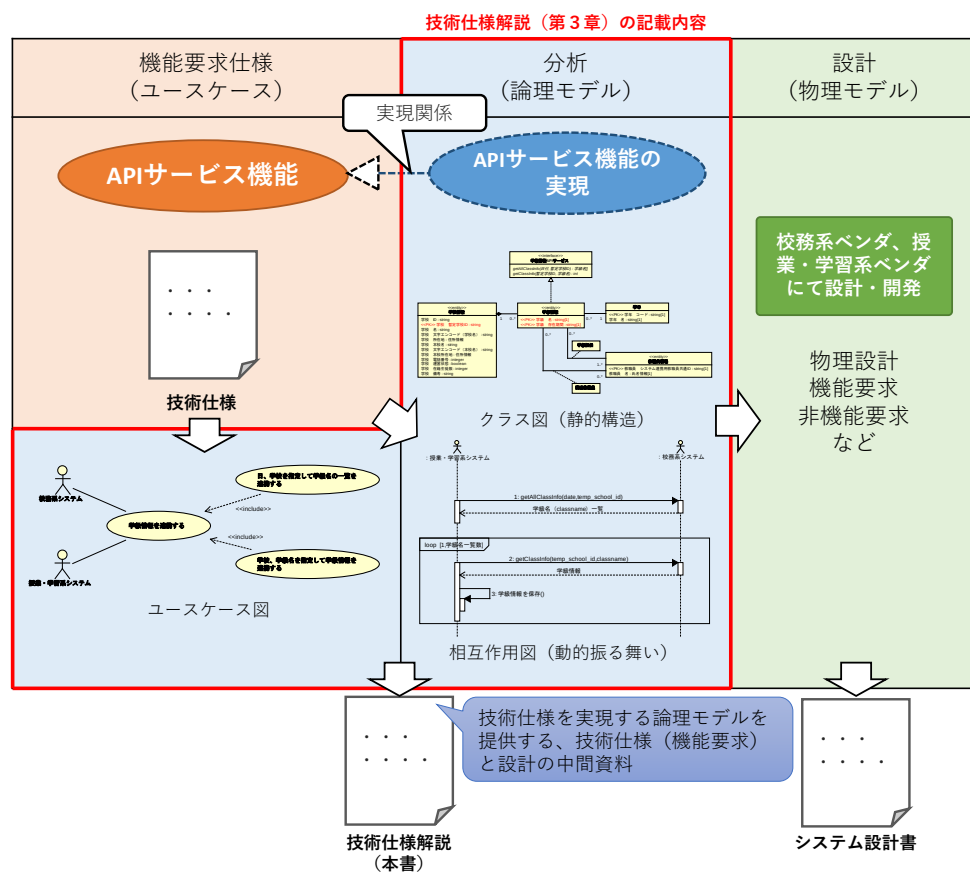


図 1-2 技術仕様解説（第3章）の内容

第4章では、スマートスクール・プラットフォームAPIサービスによるシステム間データ連携の具体的な手順およびデータ連携により実現するデータ活用事例について解説する。それぞれのデータ活用事例において必要となるAPIサービスを明確にするとともに、文部科学省「エビデンスに基づいた学校教育の改善に向けた実証事業（次世代学校支援モデル構築事業）」のデータ活用パターンとの対応関係を明確にし、システ

ム調達担当者が、校務系システム、授業・学習系システムに関する調達仕様を策定する上において、その機能要求仕様を取りまとめる参考資料という位置づけとなっている（図 1-3）。

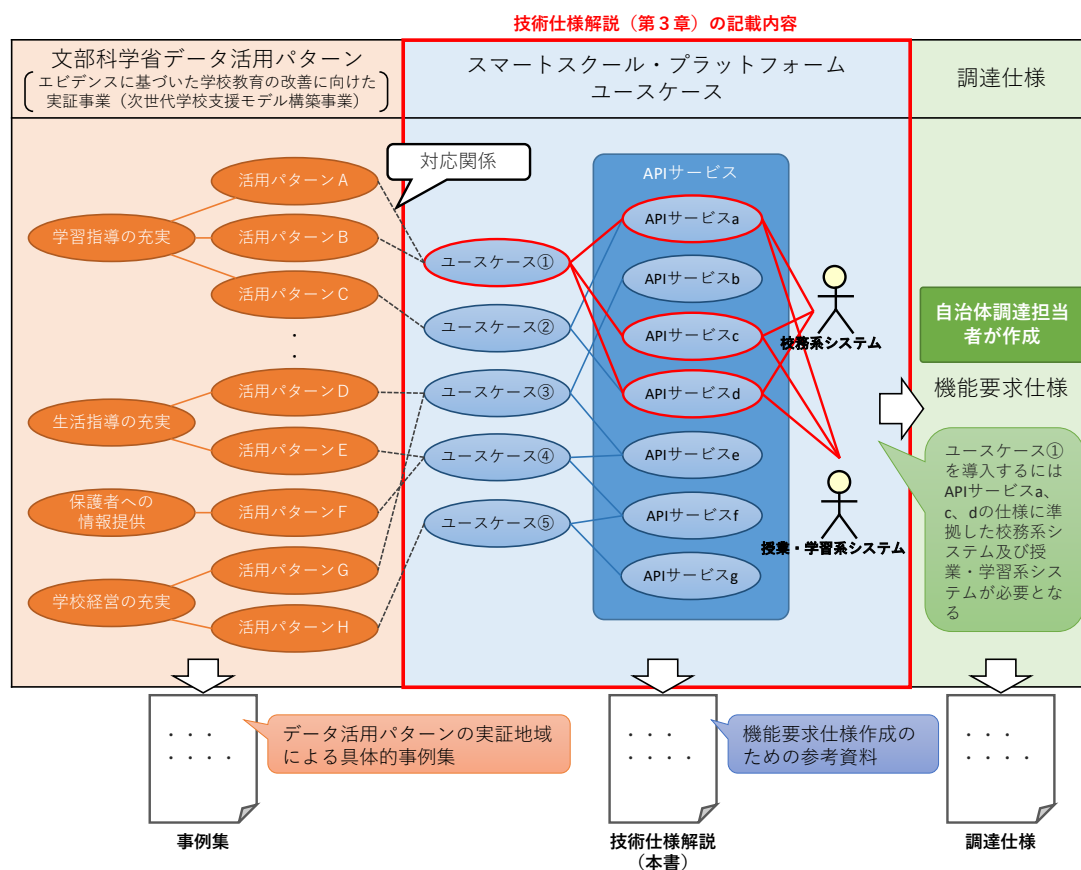


図 1-3 技術仕様解説（第4章）の内容

第1章、第2章についてはシステム開発担当者、システム調達担当者の双方を対象としている。第3章についてはシステム開発担当者を対象としており、システム調達担当者は第4章から読み進めても問題はない。

表 1-2 本書の構成と対象読者

章	内容	主な対象読者
1. はじめに	本書の目的	システム開発担当者 システム調達担当者
	本書の構成	
	本書の前提条件	
	用語集	
2. スマートスクール・プラットフォームの全体概要	概要	システム開発担当者 システム調達担当者
	目的	
	構成	
	連携対象データ	
	データ連携方法	
	情報セキュリティの確保	
	個人情報保護について	
3. API サービスの開発	(個別) 機能概要	システム開発担当者
	(個別) 静的構造	
	(個別) 動的振る舞い	
4. API サービスの利用	システム間データ連携	システム調達担当者
	データ活用事例	

巻末には参考となる補足説明、参考サイト、参考文献を記載しているので必要に応じて参照すること。

1.3 本書の前提条件

前述したように、本書は主にシステム開発担当者およびシステム調達担当者向けに、スマートスクール・プラットフォーム技術仕様について、その技術仕様を解説したものである。

本書の全般において、解説文とともに各所に統一モデリング言語である UML (Unified Modeling Language) による記法を用いているため、UML についての基礎的な知識を持っていることが前提となる。(UML については「5.3 参考文献」参照)

第3章の出欠情報 API サービス、時間割情報 API サービスにおいては、カレンダー情報に関する規定の国際標準規格である jCal、学習記録情報 API サービスにおいては、学習経験のシステム間通信に関する国際標準仕様である xAPI を用いているため、jCal、xAPI についても基礎的な知識を持っていることが前提となる。(jCal、xAPI については「5.3 参考文献」参照)

第4章においては文部科学省「エビデンスに基づいた学校教育の改善に向けた実証事業（次世代学校支援モデル構築事業）」（「5.3 参考文献」参照）のデータ活用パターン（以下、「データ活用パターン」）について、その目的、導入効果をあらかじめ把握しておくことを推奨する。(文部科学省「エビデンスに基づいた学校教育の改善に向けた実証事業（次世代学校支援モデル構築事業）」のデータ活用パターンについては「5.3 参考文献」参照)

1.4 用語集

本書で使用している用語について以下に説明する。

表 1-3 本書で使用する用語

用語	説明
校務系データ	児童生徒の成績、出欠席およびその理由、健康診断結果、指導要録、教員の個人情報など、学校が保有する情報資産のうち、それら情報を学校・学級の管理運営、学習指導、生徒指導、生活指導等に活用することを想定しており、かつ、当該情報に児童生徒がアクセスすることが想定されていない情報 (出典：文部科学省『教育情報セキュリティポリシーに関するガイドライン』) (「5.3 参考文献」参照)
授業・学習系データ	児童生徒のワークシート、作品など、学校が保有する情報資産のうち、それら情報を学校における教育活動において活用することを想定しており、かつ当該情報に教員および児童生徒がアクセスすることが想定されている情報 (出典：文部科学省『教育情報セキュリティポリシーに関するガイドライン』) (「5.3 参考文献」参照)
校務系システム	校務系ネットワーク、校務系サーバおよび校務用端末から構成される校務系データを取り扱うシステム。 (出典：『教育情報セキュリティポリシーに関するガイドライン』) (「5.3 参考文献」参照)
授業・学習系システム	学習系ネットワーク、学習系サーバ、学習者用端末および指導者用端末から構成される授業・学習系データを取り扱うシステム (出典：『教育情報セキュリティポリシーに関するガイドライン』) (「5.3 参考文献」参照)
児童生徒情報	児童生徒の氏名、性別、生年月日、住所等の基本属性情報
教職員情報	教職員の氏名および所属学校情報（学校、担任学級、担任教科、職）
所属学校情報	教職員が所属している学校、担任学級、担任教科、職。 <u>教職員は 1 つ以上の所属学校情報をもつ。所属学校情報は API 問い合わせ時点での教職員所属情報であり、過去もしくは未来の情報はもたない。</u>
学級情報	学級名および当該学級が存在した期間（年度）と当該期間における学校、学年、学級担任、関連教職員の情報
在学情報	児童生徒の学校、学年、学級の在籍履歴。 <u>児童生徒は 1 つ以上の在学情報を持つ。在学情報は、学年ごとに作成するのではなく、入学時、転入学時、編入学時、進級時、転級時、転学時に在学期間ごとに作成される。</u>
異動情報	児童生徒の入学、転入学、編入学、転学、退学、卒業、去校の異動履歴。在学情報と似ているが、同一校内での異動では作成されない点が在学情報と異なる。
出欠情報	児童生徒の日々の出欠席、遅刻、早退などの情報。 <u>指導要録上の出欠の記録（出席、欠席、出席停止・忌引等、病欠・事故欠、遅刻、早退等）に加えて、出席認定、転入前、転出後、去校後、学校休日の情報をもつ。スマート</u>

用語	説明
	スクール・プラットフォーム技術仕様ではカレンダー情報の規定に関する国際標準である jCal 形式で出欠情報を表現する。jCal の仕様では、時刻情報も含めた出欠情報の表現も可能であるが、 <u>スマートスクール・プラットフォーム技術仕様においては日単位での出欠情報を取り扱う。</u>
時間割情報	学級ごとの <u>API 問い合わせ時点での時間割情報</u> 。出欠情報と同様に、jCal 形式で時間割情報を表現する。スマートスクール・プラットフォーム技術仕様においてはモジュール授業などの分割授業を表現することもできる。
日常所見情報	スマートスクール・プラットフォーム技術仕様では、日常所見情報について、各校務系システムもしくは授業・学習系システムで記録している児童生徒に関する日常生活や学習全般の様子等を日常所見として広義の意味で捉え、日常所見の定義は、各校務系システムもしくは授業・学習系システムに依拠している。
保健室利用情報	児童生徒が保健室に来室した際の、来室日時、来室の原因および原因の発生場所、発生場面、症状等の記録情報。
学習記録情報	スマートスクール・プラットフォーム技術仕様では、児童生徒に関する単元テスト、定期テスト等のテスト・ドリルの一連の学習経験記録（学習の開始～学習の終了）およびアンケートの回答記録（アンケートの回答完了）情報としている。 <u>本書では、一連の学習経験記録情報のうち、テスト・ドリルの解答完了、単問の解答完了、およびアンケート設問の回答完了時点の学習記録情報を、特に「結果データ」と表現し、他の学習記録情報と区別している。</u>
コンテンツメタデータ	学習記録情報で取り扱うテスト・ドリル、単問、アンケートのメタデータ。スマートスクール・プラットフォーム技術仕様では、 <u>コンテンツメタデータについては、API サービスによる連携の対象外としている。</u>
Unified Modeling Language (UML)	主にオブジェクト指向分析や設計のための、記法の統一がはかられた (Unified) モデリング言語 (Modeling Language)。最新版は UML 2.5 であり、ISO/IEC 19501:2005 として UML 1.4.2 を、また、ISO/IEC 19505-1:2012 ならびに ISO/IEC 19505-2:2012 として UML 2.4.1 を標準化している。
The JSON Format for iCalendar (jCal)	カレンダー情報の規定に関する国際標準仕様。米国の任意団体 IETF によって RFC7265 として 2014 年 5 月に公開された。
Experience API(xAPI) 1.0.3	学習経験のシステム間通信に関する国際標準仕様。米国国防総省の内部組織 Advanced Distributed Learning Initiative により、1.0.3 版が 2016 年 9 月に公開された。

2. スマートスクール・プラットフォームの全体概要

本章では、スマートスクール・プラットフォームの概要、目的、構成、連携データ、連携の方法、情報セキュリティ、個人情報保護について概説する。

2.1 スマートスクール・プラットフォーム実証事業の概要

学校において、校務の情報を、学習履歴、学習記録、学習成果物等の授業・学習の記録と有効につなげ学びを「可視化」することにより、データにもとづいた児童生徒自らの学習の振り返りのみならず、教員の指導力（学習指導、生活指導等）の向上や学級・学校経営の改善、さらには、教育委員会単位の現状分析・政策立案、保護者や地域への説明等、教育の情報化の効果を飛躍的に拡大させることが期待できる。

スマートスクール・プラットフォーム実証事業では、授業・学習系システムと校務系システムとの間の、安全かつ効果的・効率的なデータ連携方法を標準化するにあたり、技術仕様の適応箇所、技術仕様のスコープ、個人情報保護、セキュリティ対策視点を考慮し、システム間のデータ連携に関わるデータ通信インタフェース仕様を策定した。

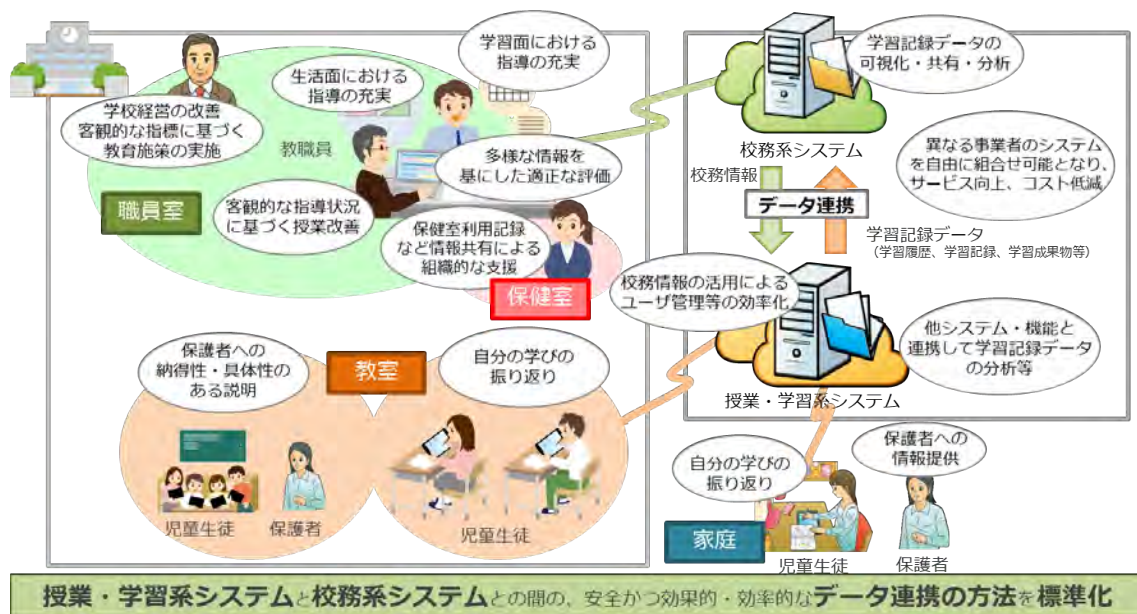


図 2-1 スマートスクール・プラットフォームの実現イメージ

2.2 スマートスクール・プラットフォームの目的

スマートスクール・プラットフォームを構築することにより、

- ① 個々の児童生徒にとっての最適な学習
- ② エビデンスベースの教育・学校経営
- ③ 学校現場における業務効率化・教職員の負担軽減
- ④ システム導入・運用コストの削減

を推進することを主な目的としている。

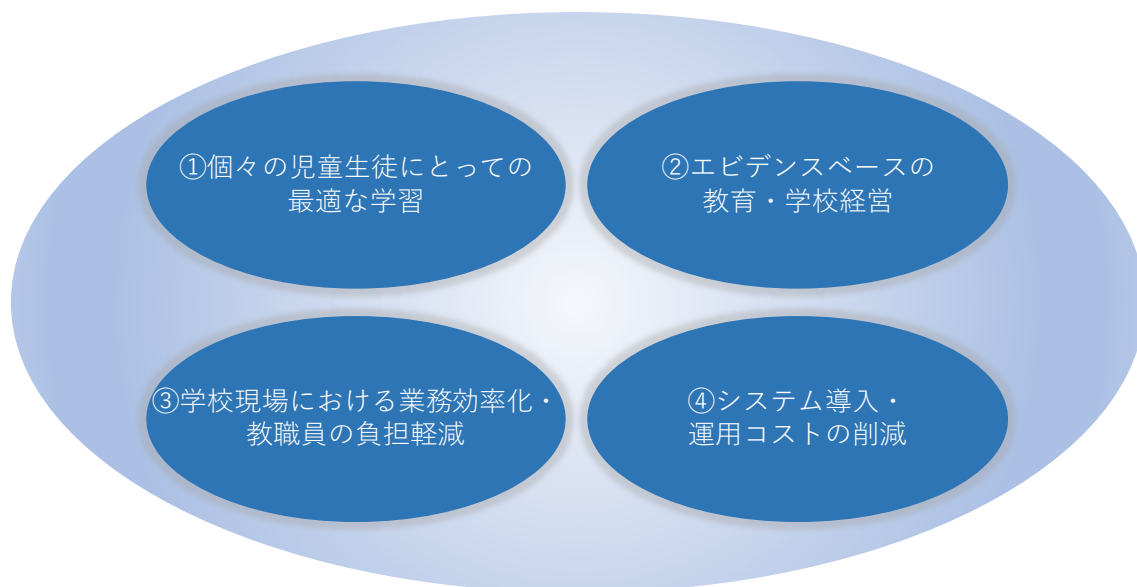


図 2-2 スマートスクール・プラットフォームの目的

昨今の統合型校務支援システムの導入促進により、従来と比較して、学校現場における校務の業務効率化や事務作業の軽減が図られてきており、今後も統合型校務支援システムの普及が進んでいくものと考えられる。この統合型校務支援システムを業務効率化・事務作業軽減だけの目的にとどめず、システム内に集約されている各種校務系データを、授業・学習系データと有効に連携させることにより、「①個々の児童生徒にとっての最適な学習」、「②エビデンスベースの教育・学校経営等による教育の質の向上」、「データの二重登録の手間を不要にすることによる③学校現場における業務効率化・負担軽減」、「インタフェース仕様の標準化による④システム導入・運用コストの削減」

等を図ることが期待される。

本技術仕様では主に、各システムに分散している校務系データおよび授業・学習系データを相互に交換するためのインタフェース仕様を規定している。

2.3 スマートスクール・プラットフォームの構成

スマートスクール・プラットフォームは、校務系データおよび授業・学習系データの、安全かつ効率的・効果的な利活用を可能にする基盤であり、校務系データおよび授業・学習系データを相互に連携する複数の校務系システムおよび授業・学習系システムから構成される。（「図 2-3 スマートスクール・プラットフォームの構成」）

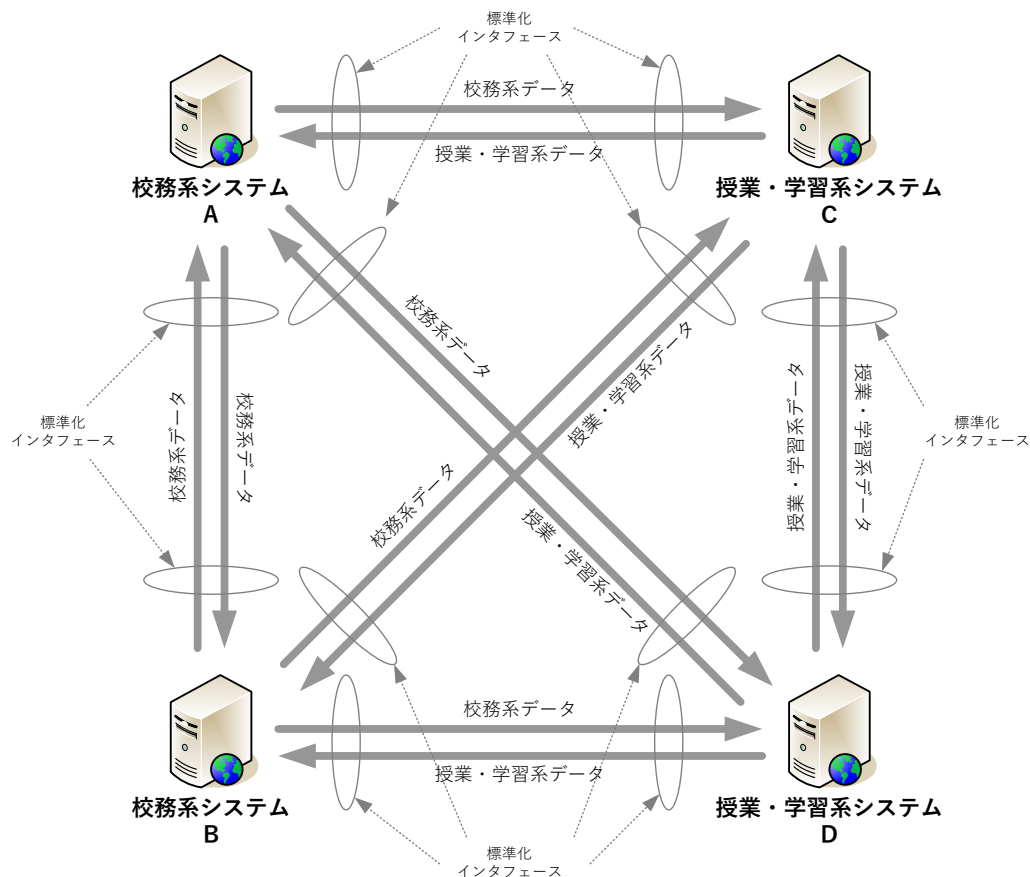


図 2-3 スマートスクール・プラットフォームの構成

スマートスクール・プラットフォームでは、データ連携する校務系システムと授業・学習系システムの組み合わせは任意である。データを受け渡す側と受け取る側のシステムが、校務系システムまたは授業・学習系システムであるかの区別はないが、本書では、便宜上、校務系システムと授業・学習系システムを明示して解説している。

2.4 連携対象とするデータについて

校務系データと授業・学習系データを効果的に連携し、活用することを目的としたシステムを、ここでは「教育データ可視化システム」と呼ぶこととする。スマートスクール・プラットフォーム技術仕様において連携対象とするデータは、教育データ可視化システムで必要となるデータで、かつ自システムにて保持していないデータを、他システムから取得する必要があるデータということになる。

スマートスクール・プラットフォーム技術仕様では、実証地域におけるデータ活用を基に、以下の校務系データ、授業・学習系データを技術仕様化の対象とした。

●個人または組織に関する基本的な属性

児童生徒情報

児童生徒の氏名、性別、生年月日、住所等の基本属性情報

教職員情報

教職員の氏名および所属学校情報（学校、担任学級、担任教科、職）

所属学校情報

教職員が所属している学校、担任学級、担任教科、職。教職員は1つ以上の所属学校情報をもつ。所属学校情報はAPI問い合わせ時点での教職員所属情報であり、過去もしくは未来の情報はもたない。

学級情報

学級名および当該学級が存在した期間（年度）と当該機関における学校、学年、学級担任、関連教職員の情報

●児童生徒の学籍に関する記録

在学情報

児童生徒の学校、学年、学級の在籍履歴。児童生徒は1つ以上の在学情報を持つ。在学情報は、学年ごとに作成するのではなく、入学時、転入時、編入時、進級時、転級時、転学（転出）時に在学期間ごとに作成される。

異動情報

児童生徒の入学、転入、編入、転学、退学、卒業、去校の異動履歴。在学情報と似ているが、同一校内での異動では作成されない点が在学情報と異なる。

●指導に関する記録

日常所見情報

スマートスクール・プラットフォーム標準仕様では、日常所見情報について、各校務系システムもしくは授業・学習系システムで記録している児童生徒に関する日常生活や学習全般の様子等を日常所見として広義の意味で捉え、日常所見の定義は、各校務系システムもしくは授業・学習系システムに依拠している。

保健室利用記録情報

児童生徒が保健室に来室した際の、来室日時、来室の原因および原因の発生場所、発生場面、症状等の記録情報。

●カレンダー型情報

出欠情報

児童生徒の日々の出欠席、遅刻、早退などの情報。指導要録上の出欠の記録（出席、欠席、出席停止・忌引等、病欠・事故欠、遅刻、早退等）に加えて、出席認定、転入前、転出後、去校後、学校休日の情報をもつ。スマートスクール・プラットフォーム標準仕様ではカレンダー情報の規定に関する国際標準であるjCal形式で出欠情報を表現する。jCalの仕様では、時刻情報も含めた出欠情報の表現も可能であるが、スマートスクール・プラットフォーム標準仕様においては日単位での出欠情報を取り扱う。

時間割情報

学級ごとのAPI問い合わせ時点での時間割情報。出欠情報と同様に、jCal形式で時間割情報を表現する。スマートスクール・プラットフォーム標準仕様においてはモジュール授業などの分割授業を表現することもできる。

●コンテンツメタデータ

学習コンテンツメタデータ

アンケートコンテンツメタデータ

学習記録情報で取り扱うテスト・ドリル、単問、アンケートのメタデータ。スマートスクール・プラットフォーム標準仕様では、コンテンツメタデータについては、APIサービスによる連携の対象外としている。

●児童生徒の学習に関する記録

学習記録情報

スマートスクール・プラットフォーム標準仕様では、児童生徒に関する単元テスト、定期テスト等のテスト・ドリルの一連の学習経験記録（学習の開始～学習の終了）およびアンケートの回答記録（アンケートの回答完了）情報としている。本書では、一連の学習経験記録情報のうち、テスト・ドリルの解答完了、単問の解答完了、およびアンケート設問の回答完了時点の学習記録情報を、特に「結果データ」として表現し、他の学習記録情報と区別している。

2.5 システム間データ連携の方法

スマートスクール・プラットフォーム技術仕様では、データ送受信の方法として、HTTPS による Web サービスを採用し、サービスモデルは REST (RESTful API) 方式としている。送受信するデータのフォーマットは、RESTful API における標準である JSON フォーマットを採用している。

RESTful API において、データ連携はデータ取得 (GET メソッド)、データ登録 (POST メソッド)、データ更新 (PUT メソッド)、データ削除 (DELETE メソッド) が代表的なデータ連携手段であるが、本技術仕様においては、データ取得 (GET メソッド) のみを規定している。

2.6 情報セキュリティの確保について

昨今、授業・学習系システムがパブリッククラウド上でサービス提供されていることを考慮すると、スマートスクール・プラットフォームを導入する上において、必然的に校務系データがインターネットを介してパブリッククラウド上に保存されることとなり、パブリッククラウドの活用を前提とした場合、情報セキュリティの確保を考慮する必要がある。

セキュリティを考慮する上において、考慮すべき点は、「情報漏洩や不正アクセスから当該情報を保護する目的である①機密性の確保」、「情報の改ざん等から保護する②完全性の確保」、「情報の紛失、破損等から保護する③可用性の確保」が挙げられる。本技術仕様においては、インタフェース標準仕様であることから、主に①の機密性の確保を目的とし、以下に挙げるセキュリティ対策を規定した。

① 閉域ネットワーク網の利用

VPN 等の閉域ネットワーク網の利用は任意とした。

② 通信の暗号化

SSL 通信、TLS1.2 以上を必須とした。

③ アクセス制御（IP 制限、ポート制限、IDS/IPS、WAF 等）

必須（ただし、技術的な方法については任意）とした。

④ アクセス制御（サーバ認証）

サーバ証明書による認証を必須とした。

本仕様は、サーバ間データ連携のインタフェース仕様であることから、サーバ間通信における情報セキュリティの要件についてのみ規定していることに留意する必要がある。スマートスクール・プラットフォームでは、クラウドサービスの利用も想定しており、クラウドサービス利用を考慮した情報セキュリティについては、文部科学省の「教育情報セキュリティポリシーに関するガイドライン」（「5.3 参考文献」参照）を検討、考慮した上で、スマートスクール・プラットフォームを導入、運用する必要がある。

2.7 個人情報保護について

個人情報の取り扱いについては、スマートスクール・プラットフォームの導入如何にかかわらず、各自治体が定める個人情報保護条例等の法令に従う必要がある。スマートスクール・プラットフォーム技術仕様では、受渡しが必須となっている項目についても、各自治体の個人情報保護条例等やポリシーに従い未設定にすることが可能な仕様としている。

3. API サービスの開発

本章では、システム開発担当者を対象に、最初に全体概要を解説したのち、個々の API サービスについて、API サービスが提供する機能、API サービスのデータモデル（静的構造）および API サービスによるシステム間データ連携方法（動的振る舞い）について解説する。

3.1 API サービスの全体概要

3.1.1 スマートスクール・プラットフォーム

スマートスクール・プラットフォームは、スマートスクール・プラットフォーム技術仕様に準拠した API サービスにより、相互に校務系データおよび授業・学習系データを連携する校務系システムおよび授業・学習系システムの複数システムのから構成される。

スマートスクール・プラットフォームの構成を図 3-1 に示す。

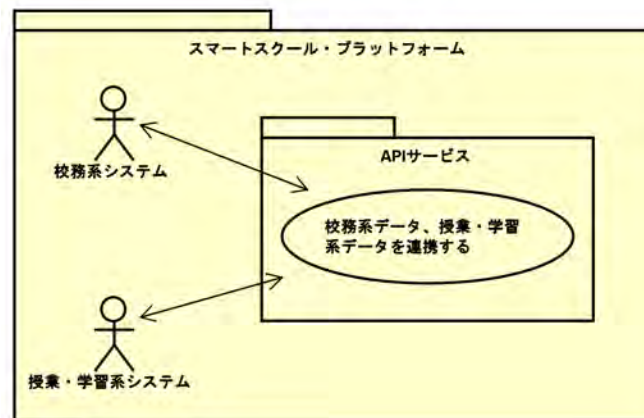


図 3-1 スマートスクール・プラットフォームの構成

各システム間では RESTful API を用いた 1 対 1（Peer to Peer）の HTTPS 通信を行う。スマートスクール・プラットフォーム技術仕様において、データ連携する校務系システムと授業・学習系システムの組み合わせは任意である。データ連携を行う上で、データを受け渡す側と、受け取る側のシステムの組み合わせは、以下のいずれかどれでも良い。（「図 3-2 データ連携システムの組み合わせ」）

- ① 「校務系システム」と「授業・学習系システム」
- ② 「校務系システム」と「校務系システム」
- ③ 「授業・学習系システム」と「授業・学習系システム」

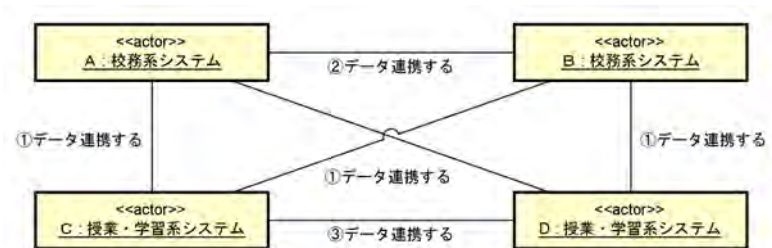


図 3-2 データ連携システムの組み合わせ

データを受け渡す側と受け取る側のシステムが、校務系システムまたは授業・学習系システムであるかの区別はないが、本書では、API サービスの解説をする上において、便宜上、校務系システムと授業・学習系システムを明示して解説している。

連携するシステムの組み合わせが任意であることと同様に、連携データの流れの方向についても、以下のいずれかどれでも良い。（「図 3-3 データ連携の方向」）

- ① 「校務系システム」から「授業・学習系システム」へデータ連携する
- ② 「授業・学習系システム」から「校務系システム」へデータ連携する
- ③ 「校務系システム」から他の「校務系システム」へデータ連携する
- ④ 「学習系システム」から他の「学習系システム」へデータ連携する

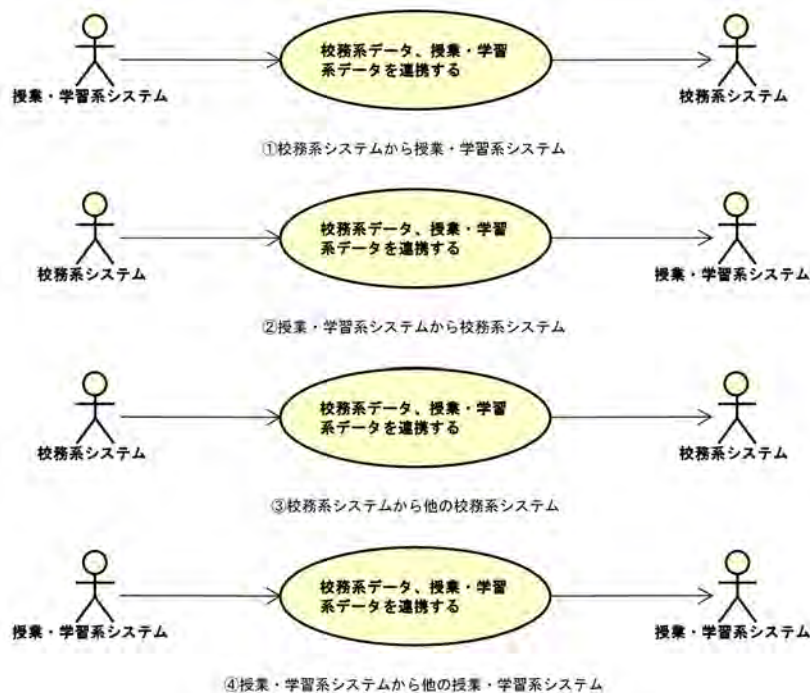


図 3-3 データ連携の方向

スマートスクール・プラットフォームでは、API サービスに直接アクセスする主体(アクター)として、技術仕様に規定するセキュリティ要件を満たした校務系システムおよび授業・学習系システムを前提としている(セキュリティ要件については「2.6 情報セキュリティの確保について」を参照)。個人利用者(教職員、児童生徒等)が、タブレット等の情報端末から直接 API サービスにアクセスすることは想定していない。(「図 3-4 API サービスのアクター」)

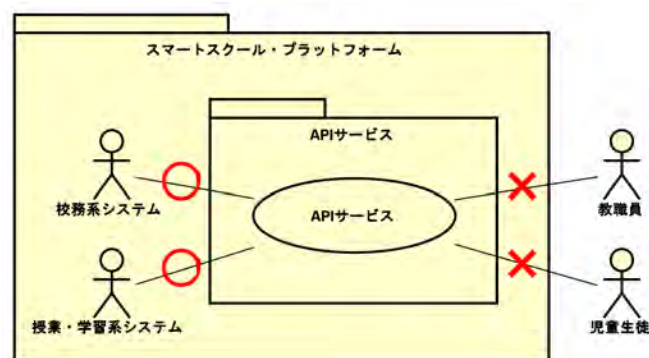


図 3-4 API サービスのアクター

スマートスクール・プラットフォームの全体機能を巻末付録の「図 5-1 全体ユース

ケース」に示した。

3.1.2 API サービスの機能

API サービスが提供する機能の全体を図 3-5 に示す。

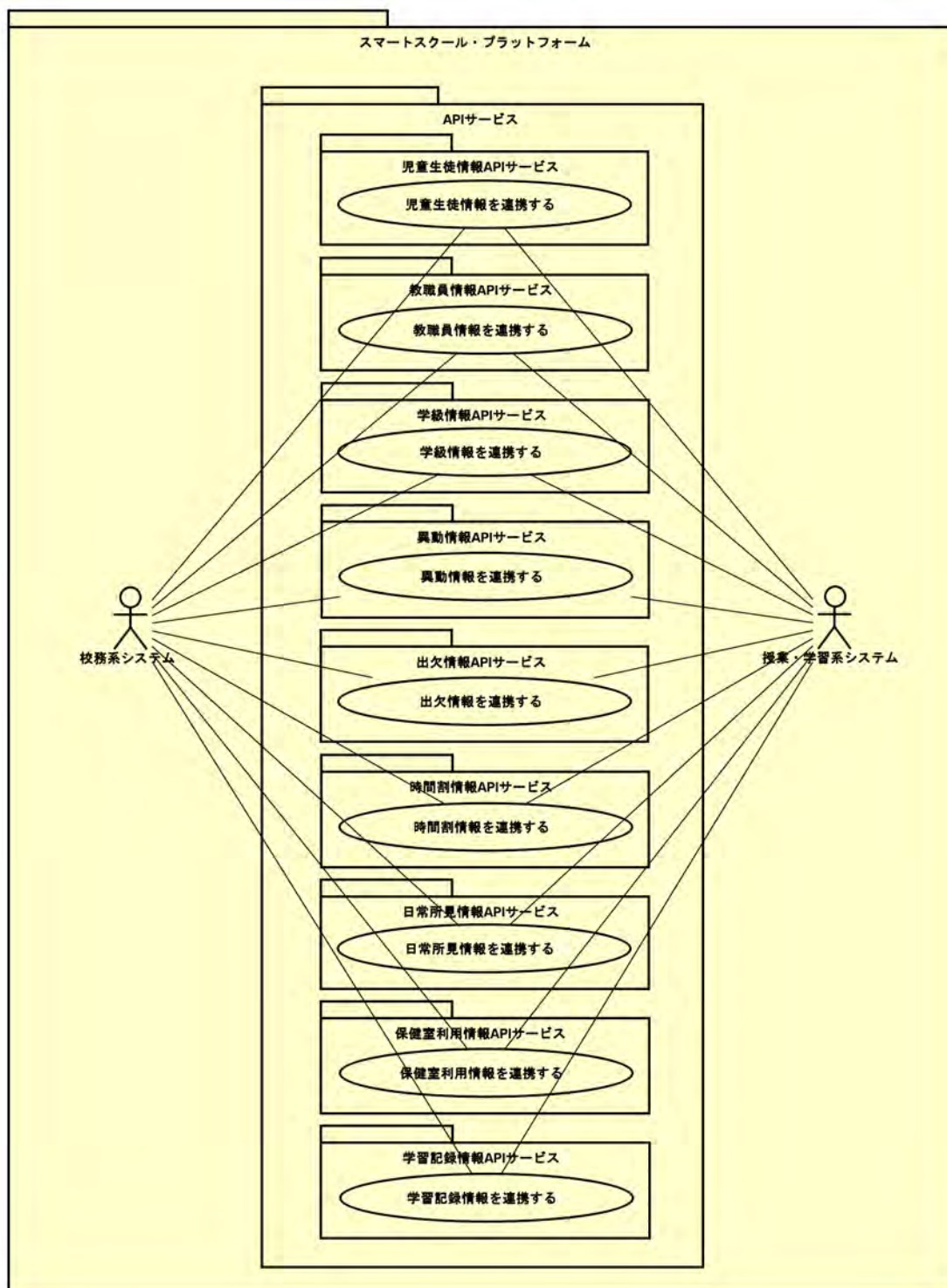


図 3-5 API サービスの全体機能

スマートスクール・プラットフォーム技術仕様 API は、以下の複数の API サービスから構成されている。

表 3-1 スマートスクール・プラットフォーム標準 API 一覧

API サービス	API	解説ページ
児童生徒情報 API サービス	getLocalStudentIds	32 頁
	getStudentInfo	
教職員情報 API サービス	getStaffIds	47 頁
	getStaffInfo	
学級情報 API サービス	getAllClassInfo	57 頁
	getClassInfo	
異動情報 API サービス	getAllTransferInfo	66 頁
	getTransferInfo	
	getAllSchoolTransferInfoList	
出欠情報 API サービス	getAttendanceRecordInfo	83 頁
時間割情報 API サービス	getTimetableInfo	94 頁
日常所見情報 API サービス	getOpinionInfo	107 頁
	getAllOpinionCodeInfo	
保健室利用情報 API サービス	getInfirmaryUseRecordInfo	119 頁
学習記録情報 API サービス	getStatementsInfoByLocalStudentId	128 頁
	getStatementsInfoWithAttachements	

各 API サービスは、1 つ、もしくは複数の API から構成される（例えば、児童生徒情報 API サービスは、指定された条件でシステム連携用児童生徒共通 ID の一覧を返す「getLocalStudentIds」、指定されたシステム連携用児童生徒共通 ID の情報を返す「getStudentInfo」の 2 つの API から構成される）。各 API サービスの詳細については「3.2 児童生徒情報 API サービス」以降で解説する。

3.1.3 API サービスのデータモデル

3.1.3.1 全体データモデル

API サービスが取り扱うデータおよび関連（データモデル）を図 3-6 に示す。

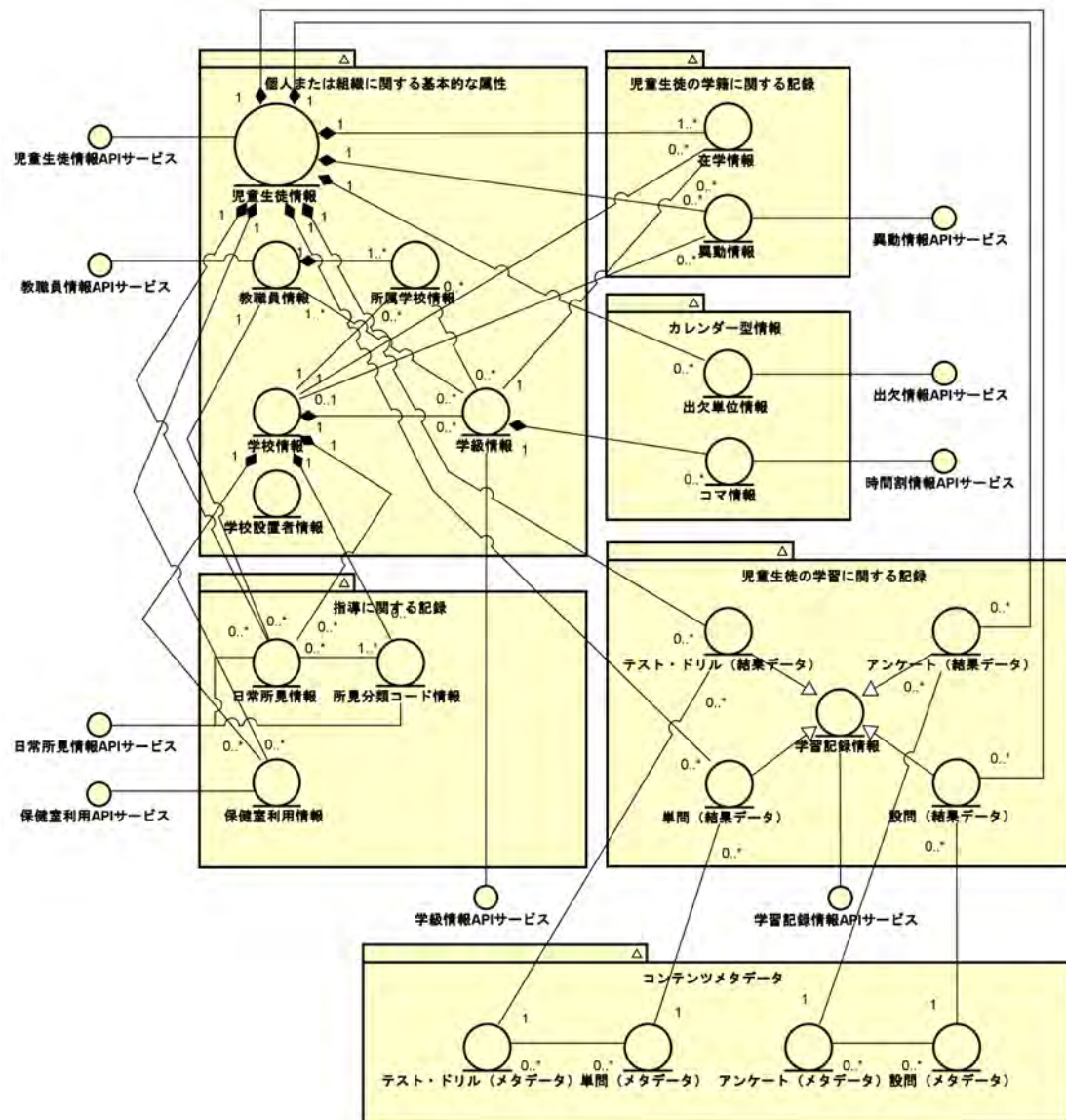


図 3-6 API サービスの全体データモデル

クラス属性も含めたデータモデルの詳細については巻末付録に記載した。（「図 5-2 全体データモデル」、「図 5-3 全体データモデル（カテゴリ別）」）

以下、各情報について概説する。

3.1.3.2 児童生徒情報

児童生徒の氏名、性別、生年月日、住所等の基本属性情報。

3.1.3.3 教職員情報

教職員の氏名および所属学校情報（学校、担任学級、担任教科、職）。

3.1.3.4 所属学校情報

教職員が所属している学校、担任学級、担任教科、職。教職員は1つ以上の所属学校情報をもつ。所属学校情報は API 問い合わせ時点での教職員所属情報であり、過去もしくは未来の情報はない。

3.1.3.5 学級情報

学級名および当該学級が存在した期間（年度）と当該期間における学校、学年、学級担任、関連教職員の情報。

3.1.3.6 在学情報

児童生徒の学校、学年、学級の在籍履歴。児童生徒は1つ以上の在学情報を持つ。在学情報は、学年ごとに作成するのではなく、入学時、転入学時、編入学時、進級時、転級時、転学時に在学期間ごとに作成される。

3.1.3.7 異動情報

児童生徒の入学、転入学、編入学、転学、退学、卒業、去校の異動履歴。在学情報と似ているが、同一校内での異動では作成されない点が在学情報と異なる。

3.1.3.8 出欠情報

児童生徒の日々の出欠席、遅刻、早退などの情報。指導要録上の出欠の記録（出席、欠席、出席停止・忌引等、病欠・事故欠、遅刻、早退等）に加えて、出席認定、転入前、転出後、去校後、学校休日の情報をもつ。スマートスクール・プラットフォーム技術仕様ではカレンダー情報の規定に関する国際標準である jCal 形式で出欠情報を表現する。jCal の仕様では、時刻情報も含めた出欠情報の表現も可能であるが、スマートスクール・プラットフォーム技術仕様においては日単位での出欠情報を取り扱う。

3.1.3.9 時間割情報

学級ごとの API 問い合わせ時点での時間割情報。出欠情報と同様に、jCal 形式で時間割情報を表現する。スマートスクール・プラットフォーム技術仕様においてはモジュール授業などの分割授業を表現することもできる。

3.1.3.10 日常所見情報

スマートスクール・プラットフォーム技術仕様では、日常所見情報について、各校務系システムもしくは授業・学習系システムで記録している児童生徒に関する日常生活や学習全般の様子等を日常所見として広義の意味で捉え、日常所見の定義は、各校務系システムもしくは授業・学習系システムに依拠している。

スマートスクール・プラットフォーム技術仕様では、所見情報として日常所見情報のみを連携対象としている。日常所見情報については各システムで独自に定義されているため、その情報として日常所見分類コード情報を連携できる仕様を規定している。

3.1.3.11 保健室利用情報

児童生徒が保健室に来室した際の、来室日時、来室の原因および原因の発生場所、発生場面、症状等の記録情報。

3.1.3.12 学習記録情報

スマートスクール・プラットフォーム技術仕様では、児童生徒に関する単元テスト、定期テスト等のテスト・ドリルの一連の学習経験記録（学習の開始～学習の終了）およびアンケートの回答記録（アンケートの回答完了）情報としている。本書では、一連の学習経験記録情報のうち、テスト・ドリルの解答完了、単問の解答完了、およびアンケート設問の回答完了時点の学習記録情報を、特に「結果データ」と表現し、他の学習記録情報と区別している。

3.1.3.13 コンテンツメタデータ

学習の結果データが学習記録情報であるのに対し、コンテンツメタデータは、学習の結果データに対応するテスト、ドリル、単問の内容、対象とする学年、教科などの情報を表す。

コンテンツメタデータについては、API サービスによる連携対象ではなく、学校情報と同様に、連携するシステム間で事前に共有しておく必要がある。

学習記録情報とコンテンツメタデータの間関係を図 3-7 に示す。詳細については、「3.10 学習記録情報」で解説する。

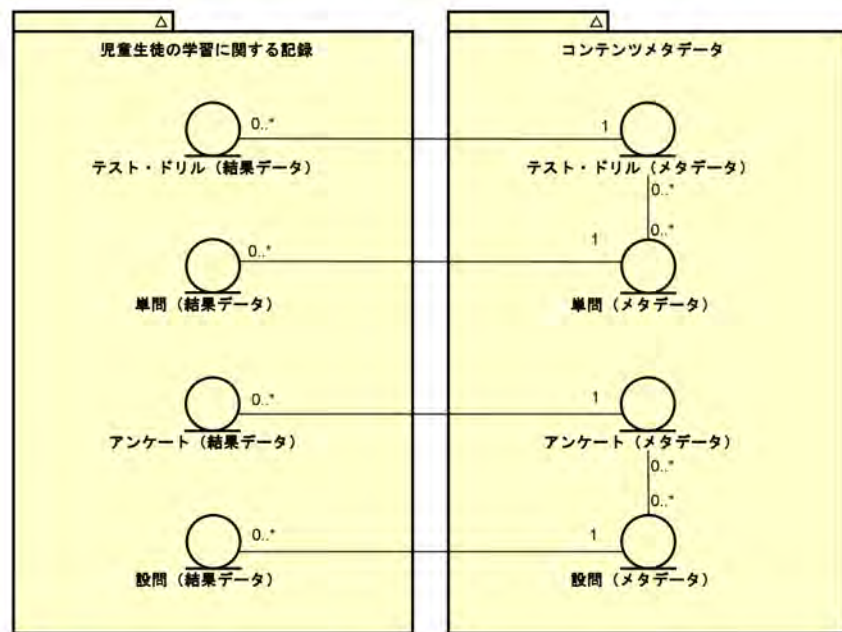


図 3-7 学習記録情報とコンテンツメタデータ

「図 3-6 API サービスの全体データモデル」では、API サービスで連携する情報と、各情報どうしの関連、多重度が記載されている。API サービスと直接の関係がない学校情報、在学情報、所属学校情報、学校設置者情報、コンテンツメタデータについては、API サービスによるデータ連携の対象外であるが、API サービスが規定されている他の情報（例えば、在学情報については児童生徒情報）をデータ連携させるために必要な情報としてデータモデルのみが規定されている。

学校情報については、API サービスによる連携対象ではないため、連携するシステム間で事前に共有しておく必要がある。

学校設置者情報についても同様にデータモデルのみの規定にとどめている。学校設置者とは、具体的には教育委員会（公立学校等）、都道府県（私立学校、特別支援学校等）などを指している。

3.1.4 共通コード

3.1.4.1 連携データとの関連

スマートスクール・プラットフォーム技術仕様では、共通コードについて標準化を図っている。スマートスクール・プラットフォーム技術仕様の共通コードを表 3-2 に示す。

表 3-2 スマートスクール・プラットフォーム標準共通コード一覧

共通コード	コード値例	使用されている情報
異動コード	入学、編入学、転入学、転学、退学等、卒業、去校	異動情報
学年コード	小学第 1 学年、学年制高等学校第 1 学年、単位制高等学校第 1 年次、他	学級情報、在学情報、異動情報、コンテンツメタデータ
校種コード	小学校、中学校、高等学校（学年制）、特別支援学校、他	学校情報
教科・科目コード	国語（小学）、社会（中学）、数学、他	所属学校情報、コンテンツメタデータ
成績コード	道徳（小学）、総合的な学習の時間（小学）、学級活動（中学）、他	
学習観点コード	国語への関心・意欲・態度、数学的な見方や考え方、他	コンテンツメタデータ
時間割コード	外国語（小学）、英語（小学）、他	コマ情報
所見分類コード		
職コード	校長、教諭、養護教諭、スクールカウンセラー、他	所属学校情報
症状コード	擦過傷、切創、頭痛、打撲、他	保健室利用情報
保健室来室原因コード	転落、転倒、精神不安、他	保健室利用情報
救急処置実施内容コード	消毒、冷却、保護者連絡、他	保健室利用情報
学校場面コード	1 時間目、登校中、給食中、自宅、他	保健室利用情報、コマ情報、学習記録情報
学校生活場所コード	普通教室、運動場、プール、自宅、他	保健室利用情報
出欠理由コード	出席、遅刻、早退、転入前、去校後、休日、病欠・事故欠、他	出欠単位情報
学習場面コード	一斉学習全般、教員による教材の提示、個別学習全般、家庭学習、他	学習記録情報

共通コードと各連携データの関連を図 3-8 に示す。

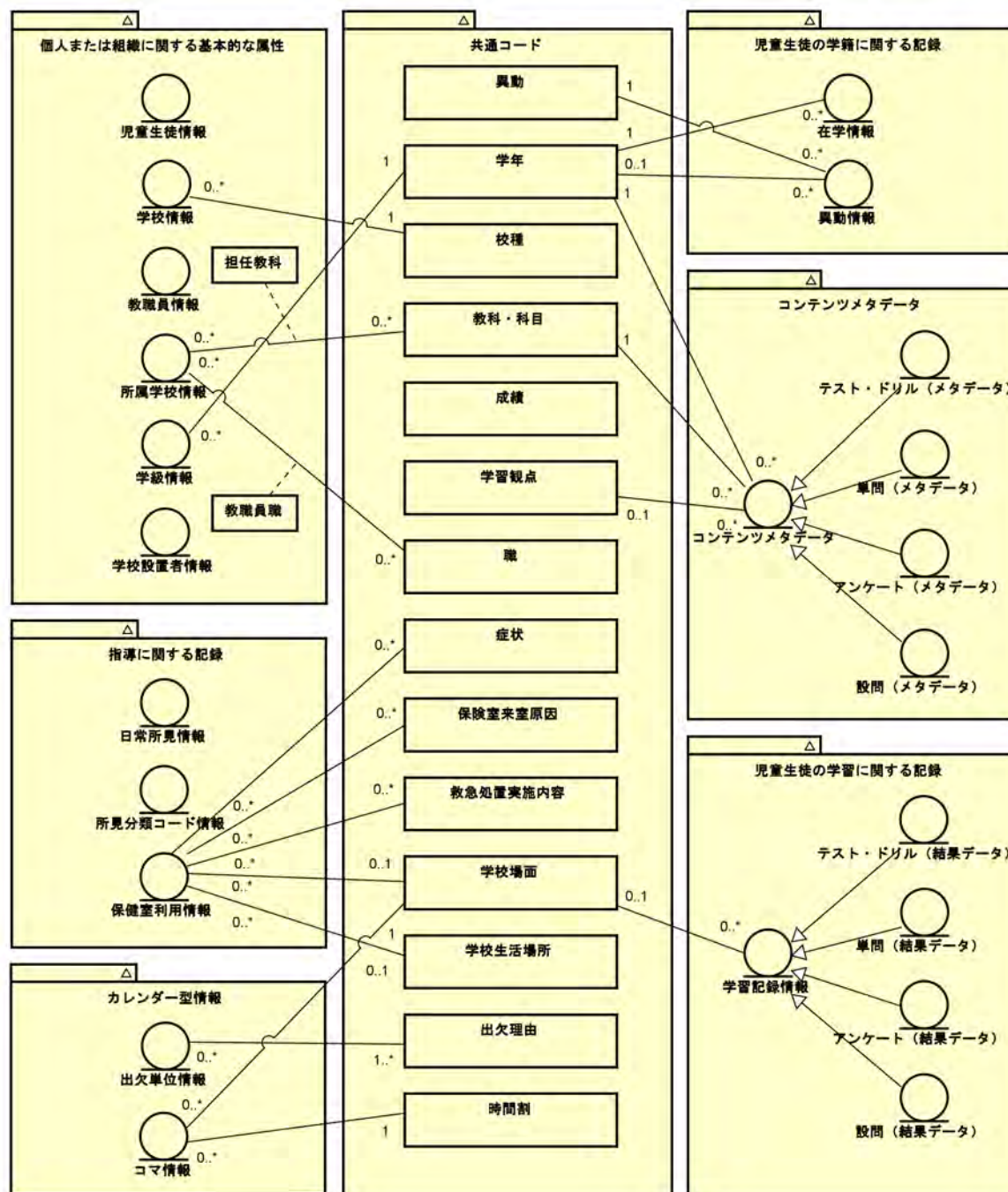


図 3-8 共通コードと連携データとの関連

成績コードについては、他のどの情報からも参照されていない。これは後述するように、成績コードが時間割コードおよび所見分類コードの一部として用いられるためである。

る。

3.1.4.2 教科・科目コードの利用範囲

教科・科目コードは、教職員情報における担当教科のほかに、成績コード、時間割コード、所見分類コードの一部としても使用される。同様に、成績コードは、時間割コード、所見分類コードの一部として、時間割コードは、所見分類コードの一部として使用される。教科・科目コード、成績コード、時間割コード、所見分類コードには、それぞれ、ユーザ独自に定義したコードを追加することができる。

成績コードは、学習指導要録上の項目（「総合的な学習の時間」、「道徳」等）が該当するが、特別な教育課程を採用している自治体等、教科にはないが成績として扱われるものをユーザ独自コードとして定義することができる。

教科・科目コード、成績コード、時間割コード、所見分類コードの関係とコード体系について、「図 3-9 教科・科目コードの適用範囲」、「表 3-3 教科・科目コードの適用範囲」に表した。

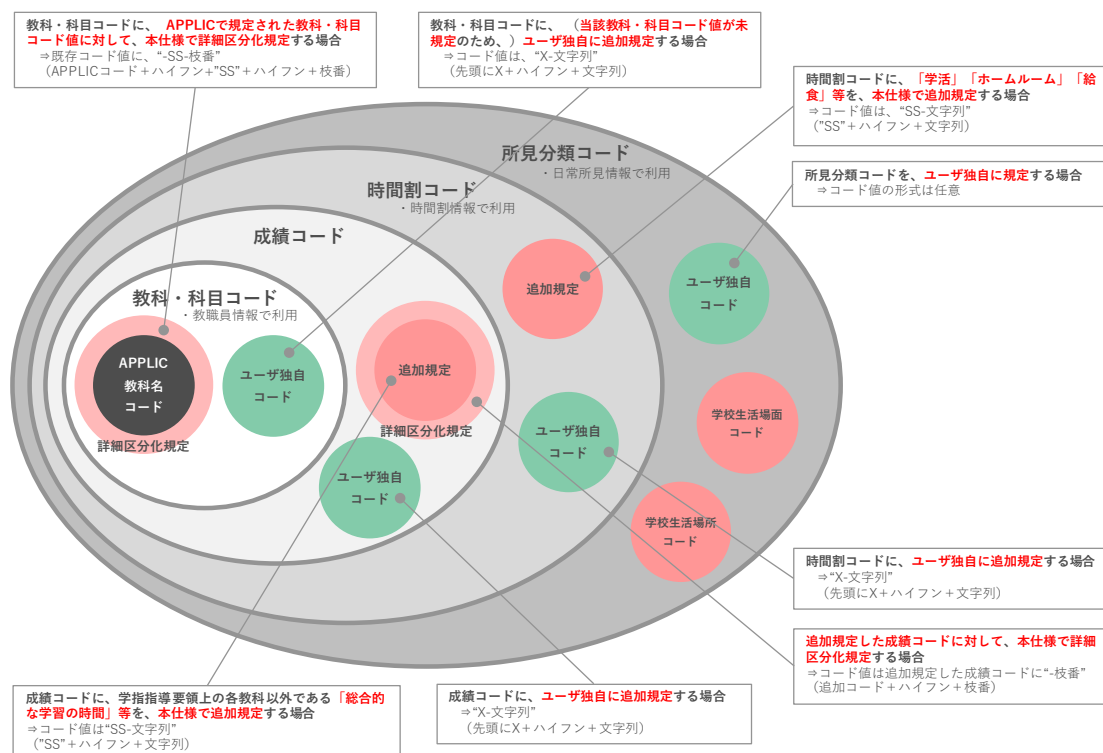


図 3-9 教科・科目コードの適用範囲

表 3-3 教科・科目コードの適用範囲

#	スマスク 共通コード 名	コードの 利用場所	ベースのコード	スマスク拡張コード	ユーザ独自コード
(1)	教科・科目コード	教職員情報	APPLIC コード (小中)教科名コード (高校)教科名コード・学科コード	APPLIC コードを詳細区分化 “-SS-枝番” (APPLIC コード+ハイフン+“SS”+枝番)	その他独自の教科・科目等 “X-文字列” (先頭に“X”+ハイフン+文字列)

(2)	成績コード	—	(1)教科・科目コード	学習指導要領記載 (指導要録参考様式記載) “SS-文字列” (“SS”+ハイフン+文字列)	その他独自の成績コード “X-文字列” (先頭に X+ハイフン+文字列)
				上記を詳細区分化 “-枝番” (上記コード+ハイフン+枝番)	

(3)	時間割コード	時間割情報	(2)成績コード	学活、ホームルーム、給食等 “SS-文字列” (“SS”+ハイフン+文字列)	その他独自の時間割選択 A・選択 B 等 “X-文字列” (先頭に X+ハイフン+文字列)
-----	--------	-------	----------	--	---

(4)	所見分類コード	日常所見情報	(3)時間割コード 学校場面コード 学校生活場所コード	—	その他独自の所見分類コード (コード値の形式は任意)
-----	---------	--------	-----------------------------------	---	-------------------------------

3.1.5 基本項目

3.1.5.1 氏名情報、住所情報

氏名情報、住所情報については基本項目として、図 3-10 のように定義している。

基本項目	
氏名情報	住所情報
氏名 外字氏名 : string[1]	住所 住所 : string[1]
氏名 文字エンコード : boolean[0..1]	住所 方書 : string[0..1]
氏名 内字氏名 : string[0..1]	住所 文字エンコード : boolean[0..1]
氏名 ふりがな : string[1]	住所 郵便番号 : integer[1]

図 3-10 基本項目（氏名情報、住所情報）

スマートスクール・プラットフォーム技術仕様では、氏名情報の「外字氏名」および住所情報の「住所」、「方書」は UTF-8 でエンコードされており、外字については Unicode 外字領域（私用領域）に定義されていることを前提とする。

3.1.5.2 文字コード

校務系システムまたは授業・学習系システムが氏名、住所を UTF-8 以外で保持している場合は、当該データを Base64 エンコードした上で、「文字エンコード」にエンコード前の文字コードを指定する。（UTF-8 の場合は、省略可）（図 3-11 参照）

渡邊 花子 : 氏名情報	渡邊 花子 : 氏名情報
氏名 外字氏名 = 渡邊 花子	氏名 外字氏名 = k27ns4FAidSOcQ==
氏名 文字エンコード = (UTF-8)	氏名 文字エンコード = shift_jis
氏名 内字氏名 = 渡邊 花子	氏名 内字氏名 = 渡邊 花子
氏名 ふりがな = わたなべ はなこ	氏名 ふりがな = わたなべ はなこ

図 3-11 文字エンコードによる設定値の違い

3.2 児童生徒情報 API サービス

3.2.1 仕様概要

3.2.1.1 getLocalStudentIds

(1) 説明

指定した期間、学校、学年、学級に 1 日でも在籍する児童生徒のシステム連携用児童生徒共通 ID（単に児童生徒 ID ともいう）の一覧を返す。

(2) パス

GET /localstudent_ids

(3) リクエストパラメータ

表 3-4 児童生徒情報 API (getLocalStudentIds) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
クエリ	since	○	期間（開始日）(YYYY-MM-DD)
クエリ	until	○	期間（終了日）(YYYY-MM-DD)
クエリ	temp_school_id	○	暫定学校 ID（単に、学校 ID ともいう）
クエリ	grade		学年コード
クエリ	class_id		学級 ID

(4) レスポンス

表 3-5 児童生徒情報 API (getLocalStudentIds) レスポンス

プロパティ名	多重度	説明
localstudent_id	0..*	システム連携用児童生徒共通 ID の配列

(5) サンプル

① リクエスト

GET /localstudent_ids?since=2018-04-01&until=2019-03-31&temp_school_id=J_0663090181&grade=J1&class_id=1%E5%B9%B4A%E7%B5%84

② レスポンス

```
{
  "results": [
    {
      "localstudent_id": [
        "S0166742",
        "S0206148",
        "S0230873",
        "S0247734",
        "S1176872"
      ]
    }
  ]
}
```

3.2.1.2 getStudentInfo

(1) 説明

指定した児童生徒の児童生徒情報を返す。

(2) パス

GET /student_info/{localstudent_id}

(3) リクエストパラメータ

表 3-6 児童生徒情報 API (getStudentInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	localstudent_id	○	システム連携用児童生徒共通 ID

(4) レスポンス

表 3-7 児童生徒情報 API (getStudentInfo) レスポンス

プロパティ名	多重度	説明
localstudent_id	1	システム連携用児童生徒共通 ID
roster_id	0..1	学籍番号
identification_name	1	住民票氏名（氏名情報）
original_name	1	外字氏名
encoding	0..1	文字エンコーディング
s13n_name	0..1	内字氏名
kana_name	1	ふりがな（平仮名）
alias_name	0..*	通称名（氏名情報）
original_name	1	外字氏名
encoding	0..1	文字エンコーディング
s13n_name	0..1	内字氏名
kana_name	1	ふりがな（平仮名）
display_name	0..*	表示名（氏名情報）
original_name	1	外字氏名
encoding	0..1	文字エンコーディング
s13n_name	0..1	内字氏名
kana_name	1	ふりがな（平仮名）
gender	1	性別（0：不明、1：男性、2：女性、9：適用不能）
birthdate	1	生年月日（YYYY-MM-DD）
regular_address	0..1	現住所（住所情報）
normal_address	1	住所
detail_address	0..1	方書
encoding	0..1	文字エンコーディング
postalcode	1	郵便番号
other_address	0..*	その他の現住所（住所情報）
normal_address	1	住所
detail_address	0..1	方書
encoding	0..1	文字エンコーディング
postalcode	1	郵便番号
nationality	0..1	国籍

(5) サンプル

① リクエスト

GET /student_info/S0000017

② レスポンス

```
{
  "results": [
    {
      "localstudent_id": "S0000017",
      "roster_id": "R33478",
      "identification_name": {
        "original_name": "八木 香菜",
        "encoding": "UTF-8",
        "s13n_name": "八木 香菜",
        "kana_name": "やぎ かな"
      },
      "alias_name": [
        {
          "original_name": "八木 香菜",
          "encoding": "UTF-8",
          "s13n_name": "八木 香菜",
          "kana_name": "やぎ かな"
        }
      ],
      "display_name": {
        "original_name": "八木 香菜",
        "encoding": "UTF-8",
        "s13n_name": "八木 香菜",
        "kana_name": "やぎ かな"
      }
    }
  ]
}
```

```
    },
    "gender": "2",
    "birthdate": "2006-07-18",
    "regular_address": {
      "normal_address": "大阪府大阪市新中央区心斎橋筋 4-12",
      "detail_address": "",
      "encoding": "UTF-8",
      "postalcode": "5420085"
    },
    "other_address": [
      {
        "normal_address": "大阪府大阪市新中央区心斎橋筋 4-12",
        "detail_address": "",
        "encoding": "UTF-8",
        "postalcode": "5420085"
      }
    ],
    "nationality": "JP"
  }
]
```


3.2.2 機能概要

児童生徒情報 API サービスのユースケース図を図 3-12 に示す。

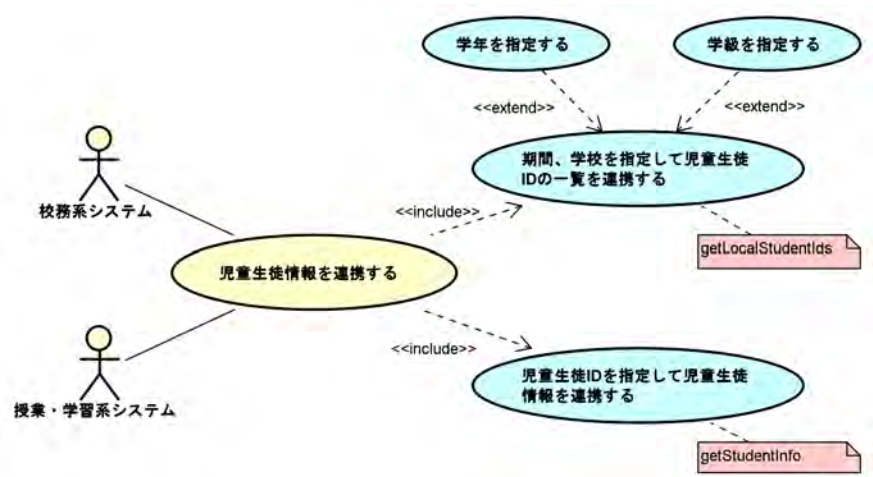


図 3-12 ユースケース図（児童生徒情報 API サービス）

API サービス `getLocalStudentIds` は、指定した期間（since、until）、暫定学校 ID（temp_school_id）に在学するすべてのシステム連携用児童生徒共通 ID（localstudent_id）の一覧を返す。指定した期間（since、until）内に 1 日でも在籍していた児童生徒が対象となる。（「図 3-13 対象となる児童生徒（児童生徒情報 API サービス）」）

	開始日 (since)	終了日 (until)	
①全期間在籍している	→	→	○対象
②期間途中に除籍した場合	→ X	→	○対象
③期間途中から在籍している場合	→	→ X	○対象
④期間中のみ在籍している場合	→ X	→	○対象
⑤期間中に在籍していない場合	→ X	→	×対象外

図 3-13 対象となる児童生徒（児童生徒情報 API サービス）

また、期間、暫定学校 ID と合わせて、学年、学級を指定することにより、特定の学級に所属する児童生徒の一覧（学級簿）を取得することができる。

API サービス `getStudentInfo` は、システム連携用児童生徒共通 ID (`localstudent_id`) を指定することにより、個々の児童生徒情報を取得することができる。この際、問い合わせ先の判断により、個人情報（氏名、通称名等）の提供が拒否される場合があることに注意する必要がある（拒否する場合は、`null` 値を設定する）。

「システム連携用児童生徒共通 ID」¹は、連携するシステム間において一意である児童生徒の識別子である。スマートスクール・プラットフォーム技術仕様では、システム連携用児童生徒共通 ID の管理・運用については規定していない。スマートスクール・プラットフォームを導入する組織（教育委員会、学校等）で管理・運用する必要がある。

¹ 本書では、「システム連携用児童生徒共通 ID」を略して「児童生徒 ID」と記載している場合もある。

3.2.3 静的構造

児童生徒情報のクラス図を図 3-14 に示す。

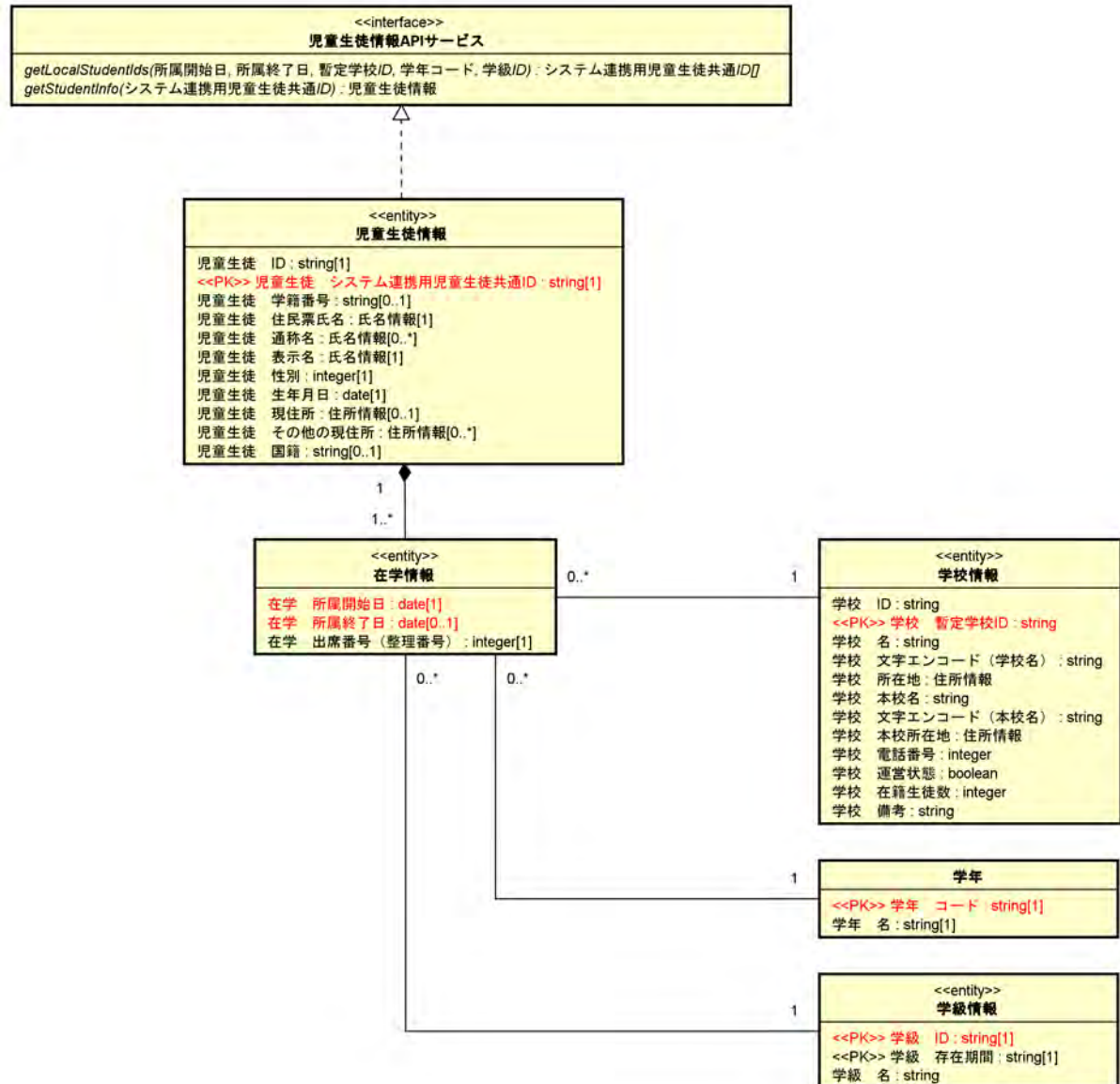


図 3-14 クラス図（児童生徒情報）

図 3-14 では、児童生徒が、どの学校、学年、学級に、どの期間在籍していたかという「在学情報」を1つ以上持つ。在学情報には所属開始日、所属終了日があり、児童生徒が進級、転級、転入学、編入学等した場合においても、API サービス getLocalStudentIds で在学情報（期間、学校、学年、学級）を参照することにより、対象となる児童生徒を取得することができる。

リクエストパラメータとクラス図との対応関係を表 3-8、表 3-9 に示す。

表 3-8 児童生徒情報 API (getLocalStudentIds) クラス図対応表

パラメータ名	クラス名	プロパティ名
since	在学情報	所属開始日
until	在学情報	所属終了日
temp_school_id	学校情報	暫定学校 ID
grade	学年	学年コード
class_id	学級	学級 ID

表 3-9 児童生徒情報 API (getStudentInfo) クラス図対応表

パラメータ名	クラス名	プロパティ名
localstudent_id	児童生徒情報	システム連携用児童生徒共通 ID

在学情報は学年（年度）ごとではなく、児童生徒の入学、転入学、編入学時および同一校内における異動（進級、転級）時に作成される。後述する「3.5 異動情報 API サービス」における異動情報は、同一校内での異動では作成されない点が在学情報と異なる。表 3-10 に在学情報と異動情報が作成されるタイミングを示す。

表 3-10 在学情報、異動情報の作成タイミング

作成タイミング	在学情報		異動情報	
入学時	○	⇒図 3-15	○	⇒図 3-33
転入学時	○		○	
編入学時	○		○	
進級時（同一校内）	○	⇒図 3-16	×	—
転級時（同一校内）	○		×	
転学（転出）時	○	⇒図 3-17	○	⇒図 3-34
退学等時	△	⇒図 3-18	○	
卒業時	△	⇒図 3-19	○	
去校時	×	—	○	

○…作成される △…更新される ×…作成されない

3.2.3.1 在学情報（入学、転入学、編入学時）

児童生徒が新年度に入学（4 月 1 日付）した場合の在学情報を図 3-15 に示す。転入学時、編入学時も同様である。

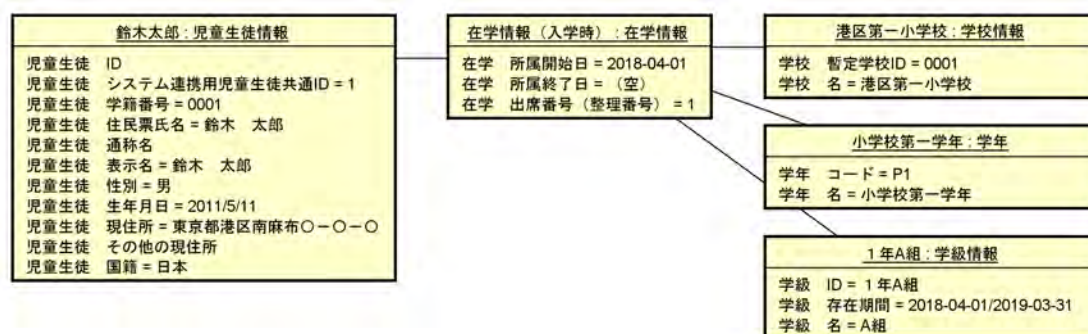


図 3-15 在学情報(入学時)

3.2.3.2 在学情報(進級、転級時)

図 3-15 に示した児童生徒が次年度に第 2 学年に進級した場合の在学情報を図 3-16 に示す。転級時も同様である。

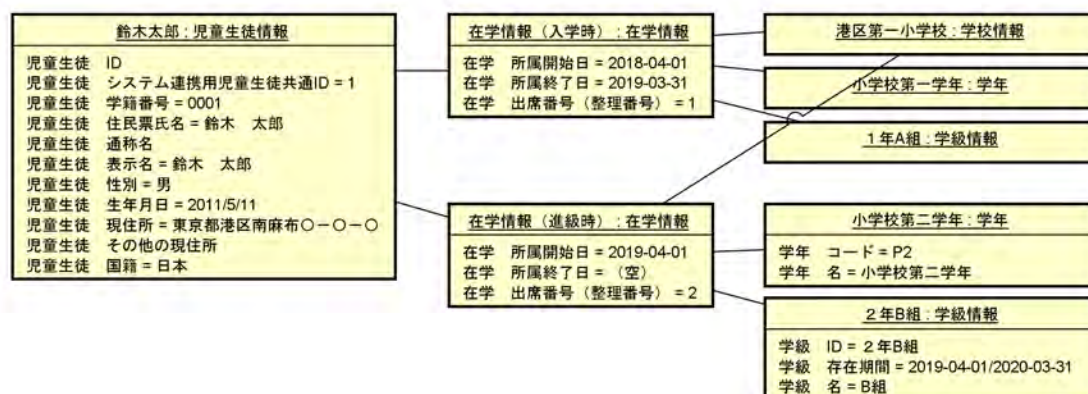


図 3-16 在学情報(進級)

3.2.3.3 在学情報(転学時)

図 3-16 で示した児童生徒が第 2 学年の途中(10 月 13 日付)で転学(転出)した場合の在学情報を図 3-17 に示す。

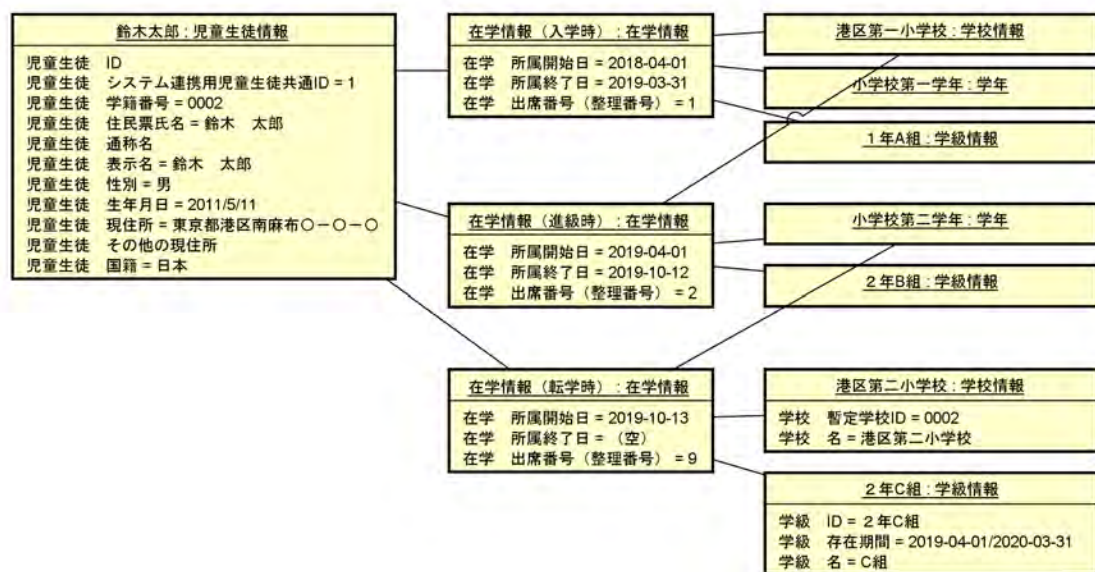


図 3-17 在学情報（転学時）

3.2.3.4 在学情報（退学等時）

図 3-16 に示した児童生徒が第 2 学年の途中（9 月 23 日付）で退学した場合の在学情報を図 3-18 に示す。

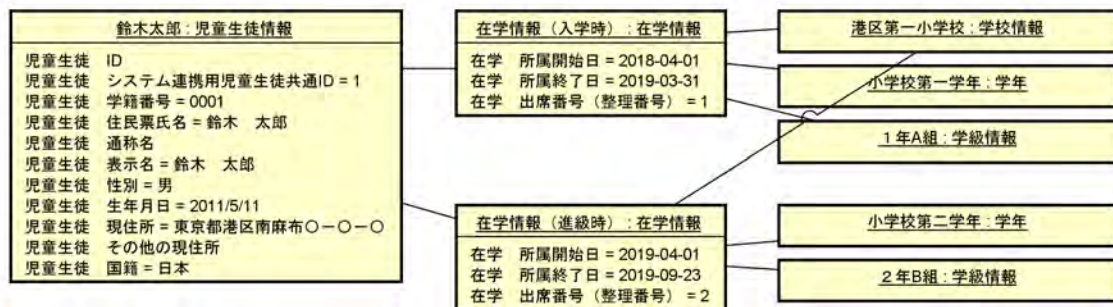


図 3-18 在学情報（退学等時）

3.2.3.5 在学情報（卒業時）

図 3-15 に示した児童生徒が第 6 学年まで進級し、卒業した場合の在学情報を図 3-19 に示す。

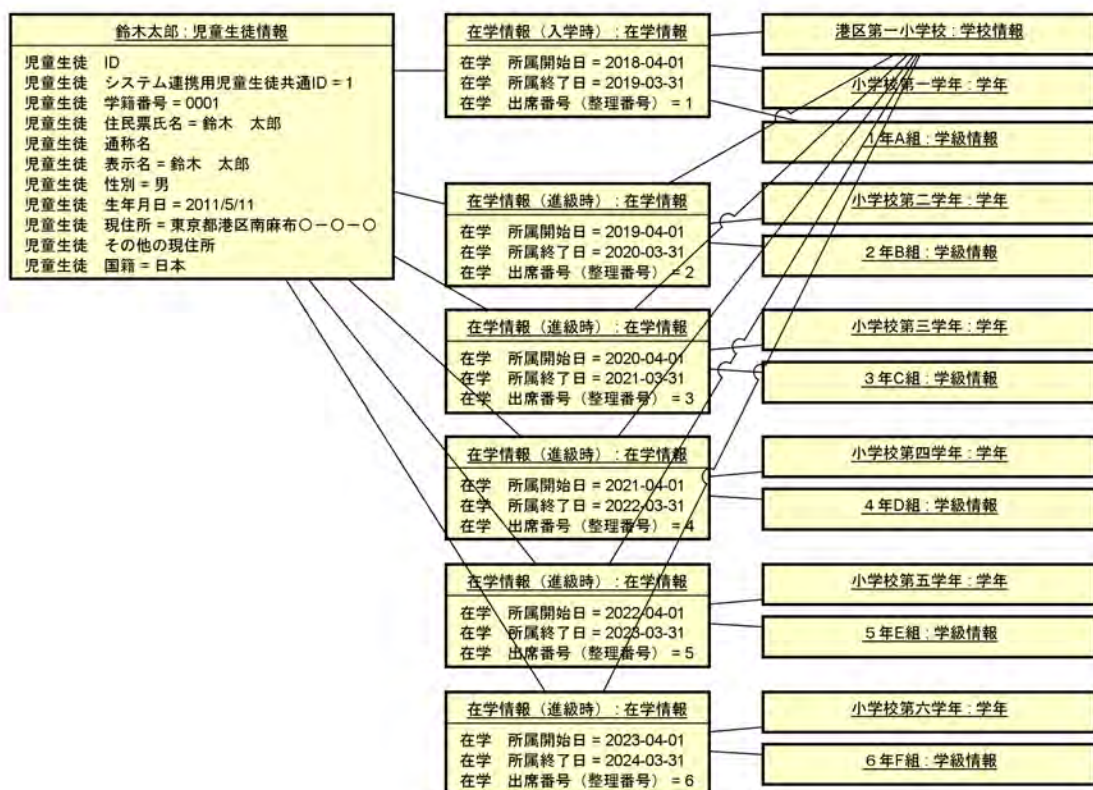


図 3-19 在学情報（卒業時）

3.2.4 動的振る舞い

児童生徒情報連携のユースケース記述を表 3-11 に示す。ここでは例として、児童生徒情報が校務系システムに登録されている前提で、それを授業・学習系システムに連携する場合について解説する。

表 3-11 ユースケース記述（児童生徒情報連携）

項目	内容
ユースケース名	児童生徒情報を連携する。
概要	校務系システムに登録されている児童生徒の属性情報（氏名、性別、生年月日、住所等）を、授業・学習系システムに連携する。
目的	システム管理業務を効率化するため。
アクター	- 授業・学習系システム - 児童生徒情報 API サービス（校務系システム） - 児童生徒情報（校務系システム）
開始条件	（ユースケースを開始する条件。開始のきっかけ・トリガー） 授業・学習系システムから校務系システムの児童生徒情報 API を呼び出す。
事前条件	（ユースケースが開始される前に満たされている条件） - 暫定学校 ID は校務系システムと授業・学習系システム間でマスタデータとして共有され、両システムに登録されていること。 - 校務系システムと授業・学習系システム間で学級情報が同期されていること。（「3.4 学級情報 API サービス」参照）
正常フロー	（正常に処理が進んだ場合） - 授業・学習系システムは、期間、学校、学年、学級を指定して（学年、学級は任意）、校務系システムにシステム連携用児童生徒共通 ID の一覧を要求する。 - 校務系システムは、指定された期間、学校、学年、学級のシステム連携用児童生徒共通 ID の一覧を抽出する。 - 校務系システムは、授業・学習系システムに、2 のシステム連携用児童生徒共通 ID の一覧を返す。 - 授業・学習系システムは、システム連携用児童生徒共通 ID を指定して、校務系システムに児童生徒情報を要求する。 - 校務系システムは、指定されたシステム連携用児童生徒共通 ID の児童生徒情報を取得する。 - 校務系システムは、授業・学習系システムに 5 の児童生徒情報を返す。 - 授業・学習系システムは、6 の児童生徒情報を自システムに登録する。次のシステム連携用児童生徒共通 ID がある場合は、4 の処理に戻る。
代替フロー	（エラーが発生した場合。代替フローを実行しても事後条件は満たされる） 上記 2 でシステム連携用児童生徒共通 ID 件数が 0 件だった場合（HTTP ステータ

項目	内容
	スコード：200)、ユースケースを終了する。
例外フロー	(ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない) 上記 5 で児童生徒情報の取得に失敗した場合 (HTTP ステータスコード：404)、ユースケースを中断する。
事後条件	(ユースケースが実行された後に満たされている条件) 授業・学習系システムと校務系システムに登録されている児童生徒情報の同期がとれていること。

児童生徒情報 API サービスのイベントフロー図を図 3-20 に示す。

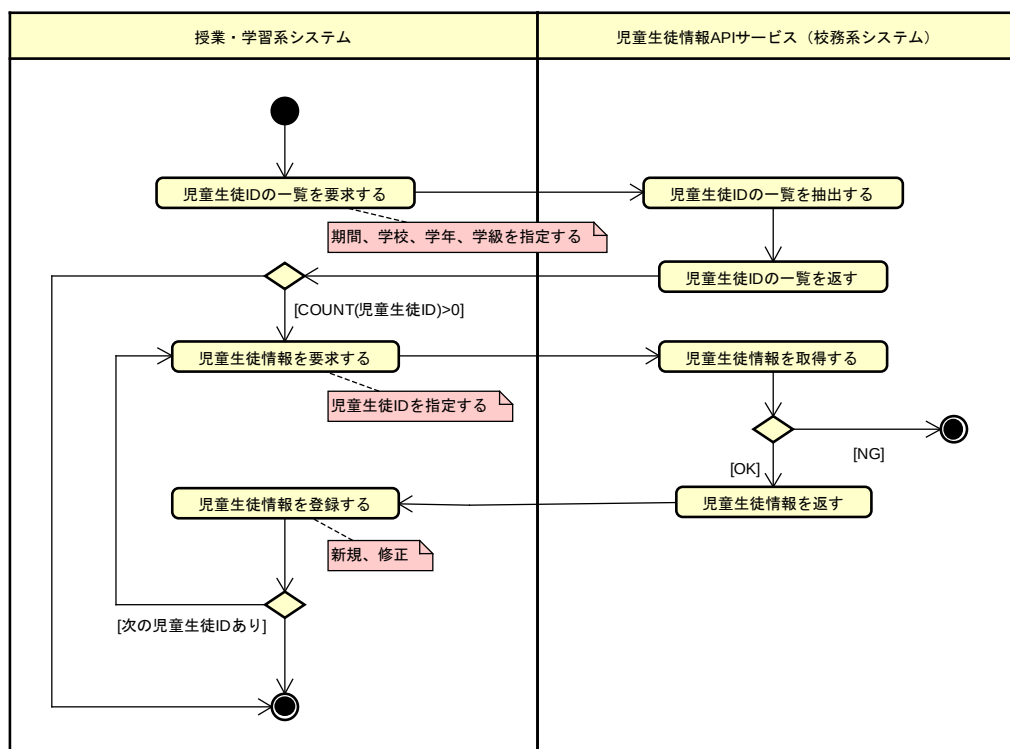


図 3-20 イベントフロー図（児童生徒情報 API サービス）

暫定学校 ID は校務系システムと授業・学習系システムとの間で事前に共有され各システムに個別に登録されているものとする。また、学級を指定して児童生徒情報を連携する場合、校務系システムと授業・学習系システムとの間で、学級情報が同期されている必要がある。（「3.4 学級情報 API サービス」参照）

児童生徒情報 API サービスのシーケンス図を図 3-21 に示す。

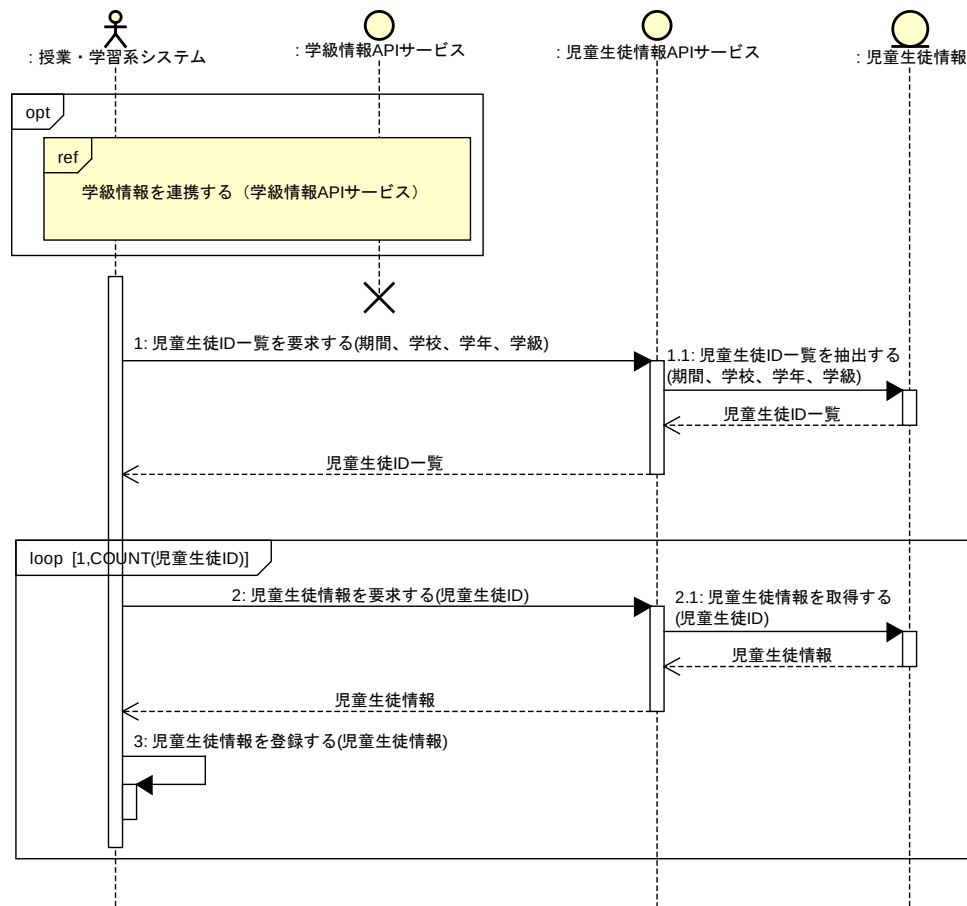


図 3-21 シーケンス図（児童生徒情報 API サービス）

最初に API サービス `getLocalStudentIds` で対象となるシステム連携用児童生徒共通 ID の一覧を取得する（図中 1）。

取得したシステム連携用児童生徒共通 ID 一覧から、システム連携用児童生徒共通 ID を指定して API サービス `getStudentInfo` を繰り返し呼び出すことにより（図中 2）、当該児童生徒情報を取得し、授業・学習系システムに登録する（図中 3）。

3.3 教職員情報 API サービス

3.3.1 仕様概要

3.3.1.1 getLocalStaffIds

(1) 説明

指定した学校、担任学級、副担任学級、担任教科の取得時点におけるシステム連携用教職員共通 ID（単に、教職員 ID ともいう）の一覧を返す。

(2) パス

GET /localstaff_ids

(3) リクエストパラメータ

表 3-12 教職員情報 API (getLocalStaffIds) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
クエリ	temp_school_id	○	暫定学校 ID
クエリ	form_class_id		担任学級 ID
クエリ	sub_form_class_id		副担任学級 ID
クエリ	subject_code		学科・教科コード

(4) レスポンス

表 3-13 教職員情報 API (getLocalStaffIds) レスポンス

プロパティ名	多重度	説明
localstaff_id	0..*	システム連携用教職員共通 ID の配列

(5) サンプル

① リクエスト

GET

/localstaff_ids?temp_school_id=J_0662711456&form_class_id=1%E5%B9%B4A%E7%B5%84

② レスポンス

```
{
  "results": [
    {
      "localstaff_id": [
        "T0002688",
        "T0004569",
        "T0011597"
      ]
    }
  ]
}
```

3.3.1.2 getStaffInfo

(1) 説明

指定した教職員の教職員情報を返す。

(2) パス

GET /staff_info/{localstaff_id}

(3) リクエストパラメータ

表 3-14 教職員情報 API (getStaffInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
クエリ	localstaff_id	○	システム連携用教職員共通 ID

(4) レスポンス

表 3-15 教職員情報 API (getStaffInfo) レスポンス

プロパティ名	多重度	説明
localstaff_id	1	システム連携用教職員共通 ID
staffname	1	教職員名（氏名情報）
original_name	1	外字氏名
encoding	0..1	文字エンコーディング
s13n_name	0..1	内字氏名
kana_name	1	ふりがな（平仮名）
belonging_schools	1..*	所属学校
temp_school_id	1	暫定学校 ID
form_class_id	0..*	担任学級 ID（例）1 年 A 組
sub_form_class_id	0..*	副担任学級 ID（例）1 年 B 組
subject_code	0..*	担任教科・学科コード
position	0..*	職コード

(5) サンプル

① リクエスト

GET /staff_info/T0008352

② レスポンス

```
{
  "results": [
    {
      "localstaff_id": "T0008352",
      "staffname": {
        "original_name": "大浦 美春",
        "encoding": "UTF-8",
        "s13n_name": "大浦 美春",
        "kana_name": "おおうら みはる"
```

```

    },
    "belonging_schools": [
      {
        "temp_school_id": "J_0663037641",
        "form_class_id": [
          "1 年 A 組"
        ],
        "sub_form_class_id": [],
        "subject_code": [
          "J090"
        ],
        "position": [
          "teacher"
        ]
      }
    ]
  }
]
}

```

3.3.2 機能概要

教職員情報 API サービスのユースケース図を図 3-22 に示す。

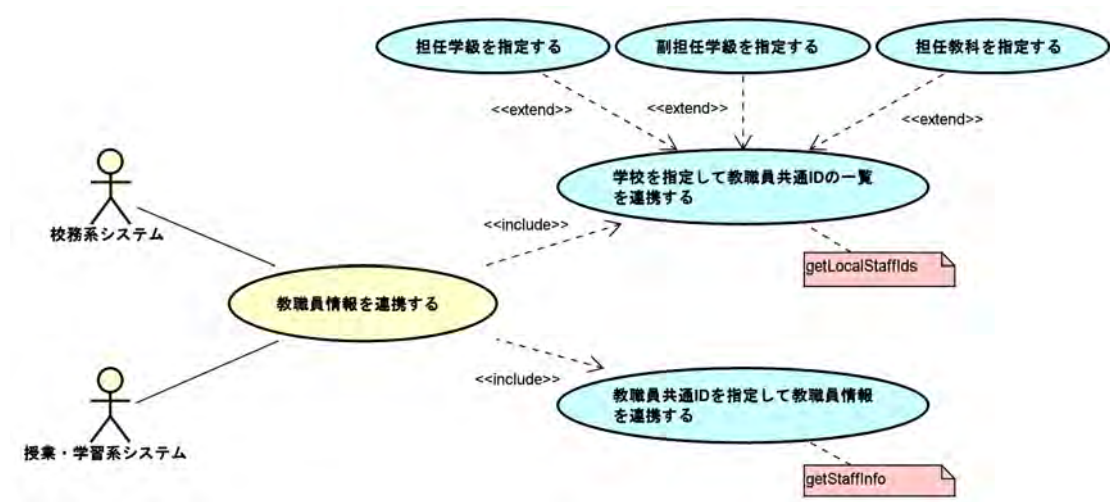


図 3-22 ユースケース図（教職員情報 API サービス）

API サービス `getLocalStaffIds` は、暫定学校 ID (`temp_school_id`) に所属するすべてのシステム連携用教職員共通 ID (`localstaff_id`) の一覧を返す。また、暫定学校 ID と合わせて、担任学級、副担任学級、担任教科を指定することにより、特定の学級、教科を担当する教職員共通 ID の一覧を取得することもできる。

API サービス `getStaffInfo` は、システム連携用教職員共通 ID (`localstaff_id`) を指定することにより、特定の教職員情報を取得することができる。

API サービス `getLocalStaffIds` および `getStaffInfo` で取得できる教職員情報は、API 問い合わせ時点における教職員情報である。過去および未来の時点における教職員情報は取得できないことに注意する必要がある。

「システム連携用教職員共通 ID」は、連携するシステム間において一意である教職員の識別子である。スマートスクール・プラットフォーム技術仕様では、システム連携用教職員共通 ID の管理・運用については規定していない。スマートスクール・プラットフォームを導入する組織（教育委員会、学校等）で管理・運用する必要がある。

3.3.3 静的構造

教職員情報のクラス図を図 3-23 に示す。

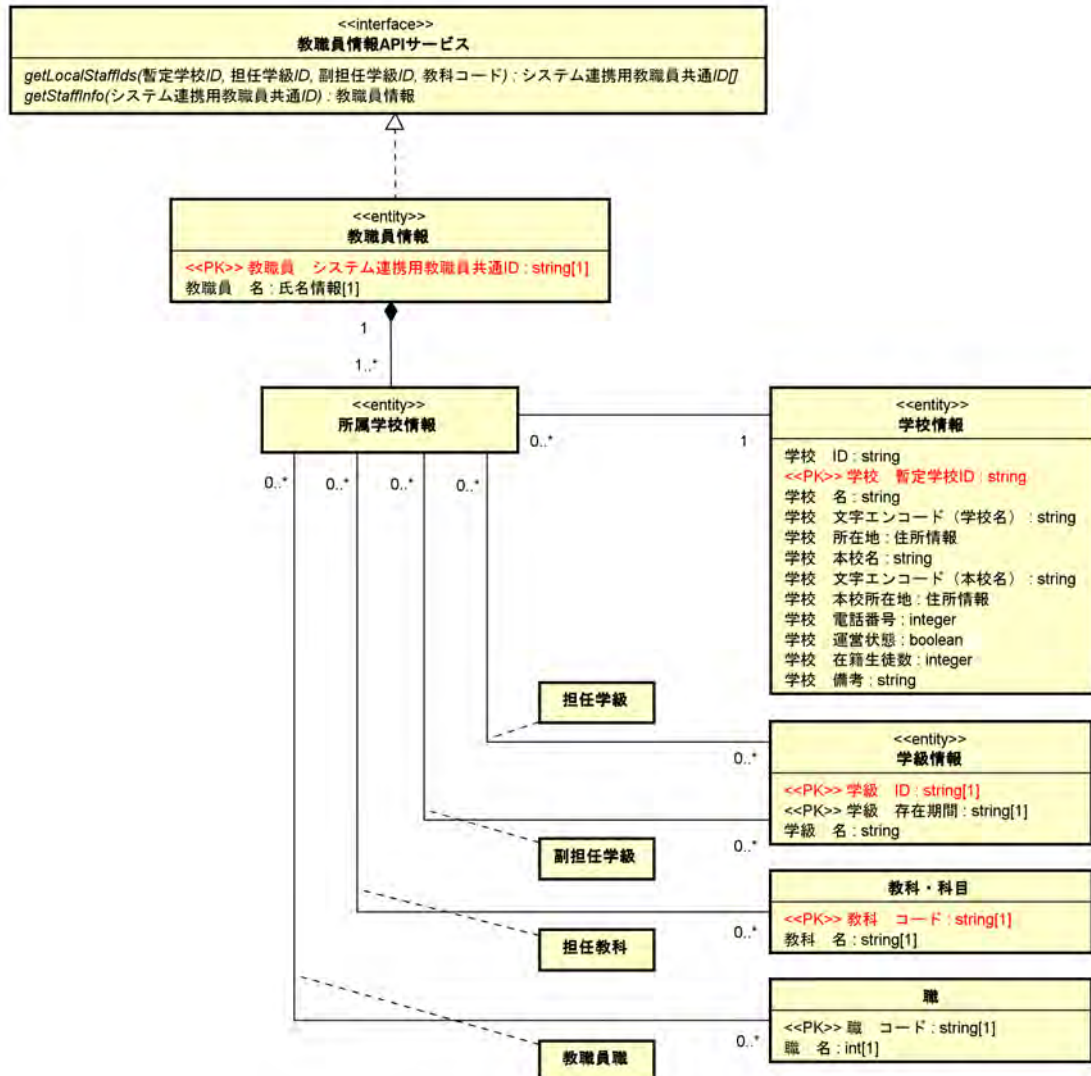


図 3-23 クラス図（教職員情報 API サービス）

図 3-23 では、教職員が、どの学校、学級、教科を担当していたかという「所属学校情報」を1つ以上持つ。教職員が所属学校情報を複数持つことにより、API サービス getLocalStaffIds で教職員の所属学校情報（学校、担任学級、副担任学級、担当教科）を参照することにより、対象となる教職員の一覧を取得することができる。この所属学校情報には期間の情報はないため、前述したように、教職員情報は API 問い合わせ時点での情報であることに注意する必要がある。

リクエストパラメータとクラス図との対応関係を表 3-16、表 3-17 に示す。

表 3-16 教職員情報 API (getLocalStaffIds) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID
form_class_id	学級情報 (担任学級)	学級 ID
sub_form_class_id	学級情報 (副担任学級)	学級 ID
subject_code	職	職コード

表 3-17 教職員情報 API (getStaffInfo) クラス図対応表

パラメータ名	クラス名	プロパティ名
localstaff_id	教職員情報	システム連携用教職員共通 ID

3.3.4 動的振る舞い

教職員情報連携のユースケース記述を表 3-18 に示す。ここでは例として、教職員情報が校務系システムに登録されている前提で、それを授業・学習系システムに連携する場合について解説する。

表 3-18 ユースケース記述 (教職員情報連携)

項目	内容
ユースケース名	教職員情報を連携する。
概要	校務系システムに登録されている教職員の属性情報 (氏名) と所属情報 (学校、担任学級、副担任学級、担任教科、職) を、授業・学習系システムに連携する。
目的	システム管理業務を効率化するため。
アクター	- 授業・学習系システム - 教職員情報 API サービス (校務系システム) - 教職員情報 (校務系システム)
開始条件	(ユースケースを開始する条件。開始のきっかけ・トリガー) 授業・学習系システムから校務系システムの教職員情報 API を呼び出す。
事前条件	(ユースケースが開始される前に満たされている条件) - 暫定学校 ID は校務系システムと授業・学習系システム間でマスタデータとして共有され、両システムに登録されていること。 - 校務系システムと授業・学習系システム間で学級情報が同期されていること。(「3.4 学級情報 API サービス」参照)
正常フロー	(正常に処理が進んだ場合)

項目	内容
	1. 授業・学習系システムは、学校、担任学級、副担任学級、担任教科を指定して（担任学級、副担任学級、担任教科は任意）、校務系システムにシステム連携用教職員共通 ID の一覧を要求する。 2. 校務系システムは、指定された学校、担任学級、副担任学級、担任教科のシステム連携用教職員共通 ID の一覧を抽出する。 3. 校務系システムは、授業・学習系システムに、2 のシステム連携用教職員共通 ID の一覧を返す。 4. 授業・学習系システムは、システム連携用教職員共通 ID を指定して、校務系システムに教職員情報を要求する。 5. 校務系システムは、指定されたシステム連携用教職員共通 ID の教職員情報を取得する。 6. 校務系システムは、授業・学習系システムに 5 の教職員情報を返す。 7. 授業・学習系システムは、6 の教職員情報を自システムに登録する。次のシステム連携用教職員共通 ID がある場合は、4 の処理に戻る。
代替フロー	（エラーが発生した場合。代替フローを実行しても事後条件は満たされる） 上記 2 でシステム連携用教職員共通 ID 件数が 0 件だった場合（HTTP ステータスコード：200）、ユースケースを終了する。
例外フロー	（ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない） 上記 5 で教職員情報の取得に失敗した場合（HTTP ステータスコード：404）、ユースケースを中断する。
事後条件	（ユースケースが実行された後に満たされている条件） 授業・学習系システムと校務系システムに登録されている教職員情報の同期がとれていること。

教職員情報 API サービスのイベントフロー図を図 3-24 に示す。

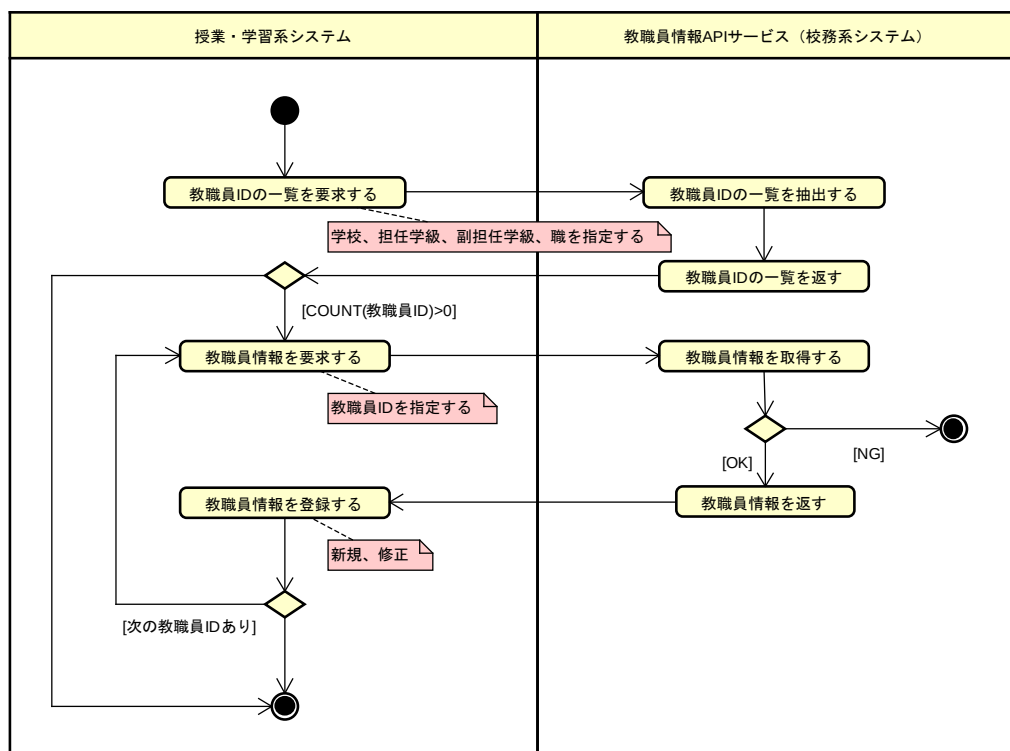


図 3-24 イベントフロー図（教職員情報 API サービス）

暫定学校 ID は校務系システムと授業・学習系システムとの間で事前に共有されているものとする。また、担任学級、副担任学級を指定して教職員情報を連携する場合、校務系システムと授業・学習系システムとの間で学級情報が同期されている必要がある。（「3.4 学級情報 API サービス」参照）

教職員情報 API サービスのシーケンス図を図 3-25 に示す。

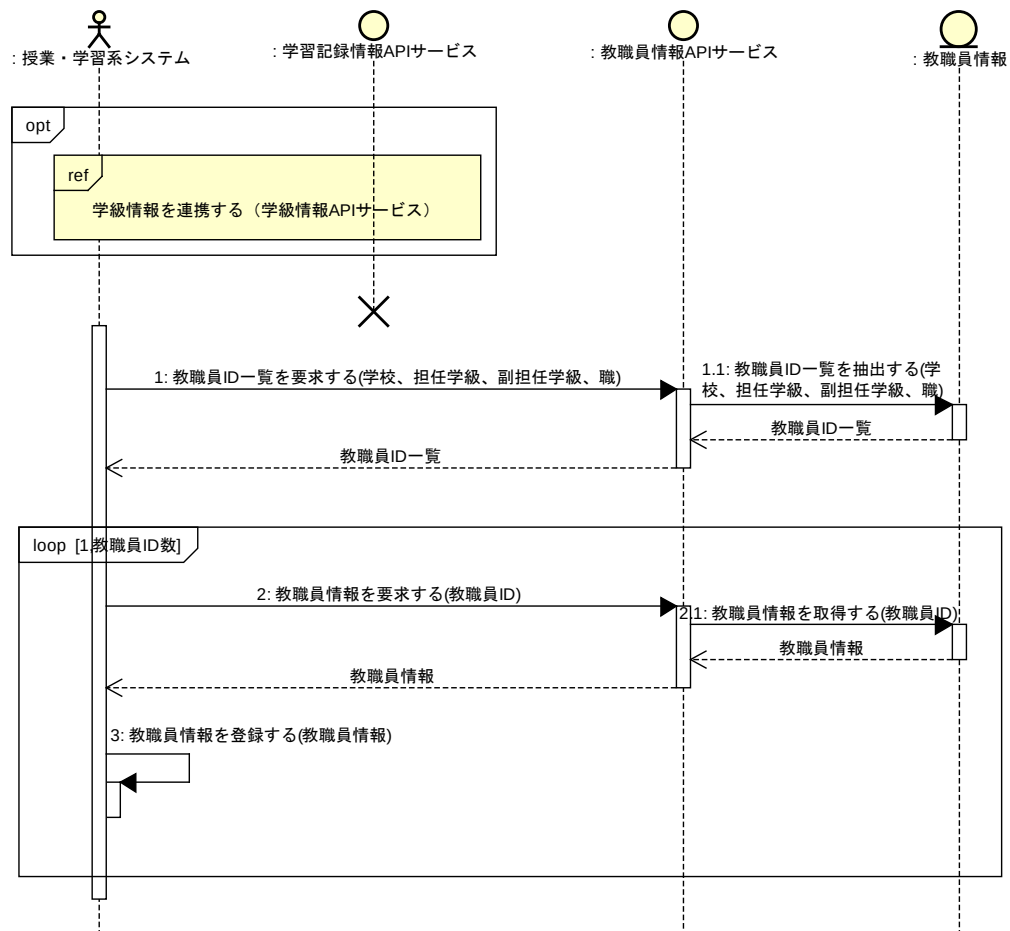


図 3-25 シーケンス図（教職員情報 API サービス）

最初に API サービス `getLocalStaffIds` で対象となるシステム連携用教職員共通 ID の一覧を取得する（図中 1）。

取得したシステム連携用教職員共通 ID 一覧から、システム連携用教職員共通 ID を指定して API サービス `getStaffInfo` を繰り返し呼び出すことにより（図中 2）、当該教職員情報を取得し、授業・学習系システムに登録する（図中 3）。

3.4 学級情報 API サービス

3.4.1 仕様概要

3.4.1.1 getAllClassInfo

(1) 説明

指定した学校、日付時点の学級 ID の一覧を返す。

(2) パス

GET /class_info/{temp_school_id}

(3) リクエストパラメータ

表 3-19 学級情報 API (getAllClassInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	temp_school_id	○	暫定学校 ID
クエリ	date	○	日付 (YYYY-MM-DD)

(4) レスポンス

表 3-20 学級情報 API (getAllClassInfo) レスポンス

プロパティ名	多重度	説明
class_id	0..*	学級 ID の配列

(5) サンプル

① リクエスト

GET /class_info/J_0662711456?date=2018-04-01

② レスポンス

```
{  
  "results": [  
    {
```

```

        "class_id": [
            "1 年 A 組",
            "1 年 B 組",
            "1 年 C 組",
            "1 年 D 組",
            "1 年 E 組",
            "2 年 A 組",
            "2 年 B 組",
            "2 年 C 組",
            "2 年 D 組",
            "2 年 E 組",
            "3 年 A 組",
            "3 年 B 組",
            "3 年 C 組",
            "3 年 D 組",
            "3 年 E 組"
        ]
    }
]
}

```

3.4.1.2 getClassInfo

(1) 説明

指定した学校、学級、日付時点の学級情報を返す。

(2) パス

GET /class_info/{temp_school_id}/{class_id}

(3) リクエストパラメータ

表 3-21 学級情報 API (getClassInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	temp_school_id	○	暫定学校 ID
パス	class_id	○	学級 ID
クエリ	date	○	日付 (YYYY-MM-DD)

(4) レスポンス

表 3-22 学級情報 API (getClassInfo) レスポンス

プロパティ名	多重度	説明
class_id	1	学級 ID
term	1	存在期間 (YYYY-MM-DD/YYYY-MM-DD)
grade	1	学年コード
classname	1	学級名 (例) 1 組、A 組
form_teacher_id	1..*	担任教職員 ID
related_staff_id	0..*	関連教職員 ID

(5) サンプル

① リクエスト

GET /class_info/J_0662711456/1%E5%B9%B4A%E7%B5%84?date=2018-04-01

② レスポンス

```
{
  "results": [
    {
      "class_id": "1 年 A 組",
      "term": "2018-04-01/2019-03-31",
      "grade": "J1",
      "classname": "A 組",
      "form_teacher_id": [
        "T0004569"
      ]
    }
  ]
}
```

```
    ],  
    "related_staff_id": [  
        "T0013937"  
    ]  
}  
]  
}
```


3.4.2 機能概要

学級情報 API サービスのユースケース図を図 3-26 に示す。

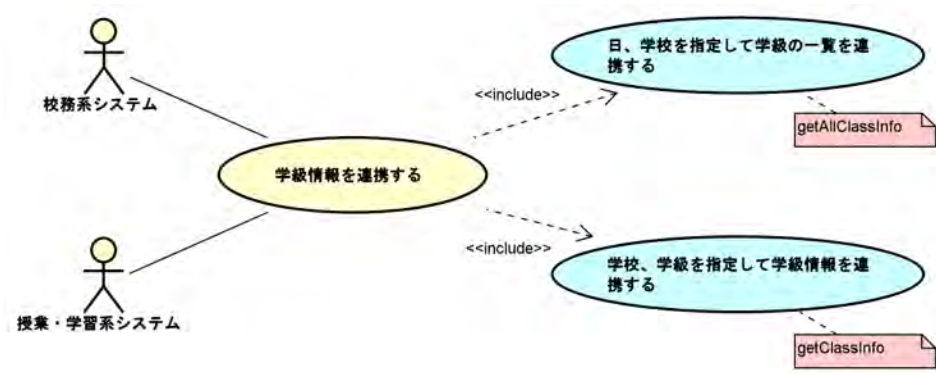


図 3-26 ユースケース図（学級情報 API サービス）

API サービス `getAllClassInfo` は、指定した日（date）に、指定した暫定学校 ID（temp_school_id）に存在するすべての学級 ID（class_id）の一覧を返す。

API サービス `getClassInfo` で、指定した暫定学校 ID（temp_school_id）、学級 ID（class_id）の学級情報を取得することができる。

3.4.3 静的構造

学級情報のクラス図を図 3-27 に示す。

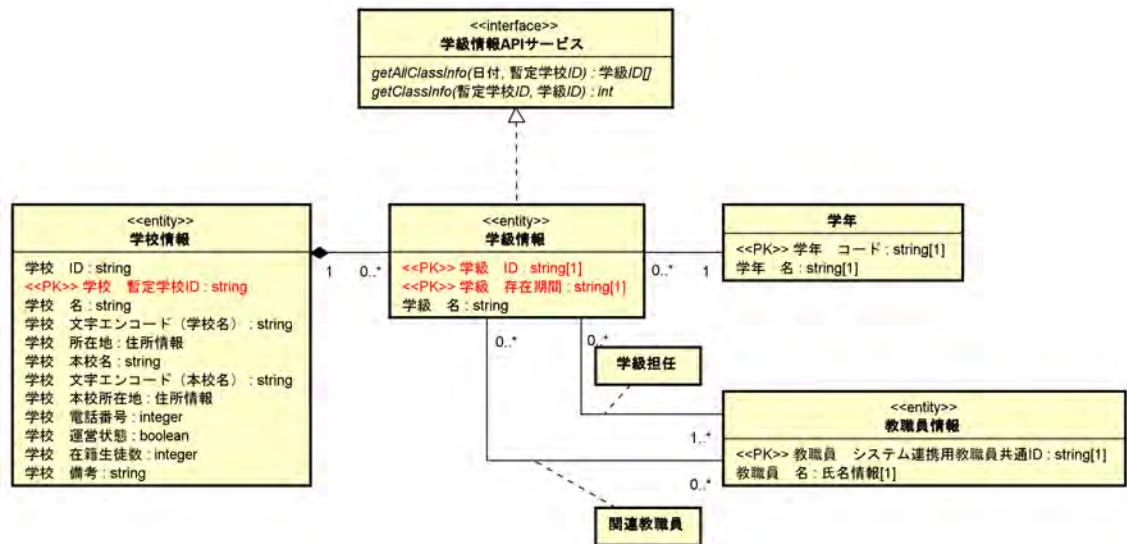


図 3-27 クラス図（学級情報 API サービス）

図 3-27 では、学級が、どの学校に所属しているかの「学校情報」を持つ。API サービス `getAllClassInfo` で学級が所属する学校を参照することにより、対象となる学級の一覧を取得することができる。

学級情報において、学年を定義できない複式学級および特別支援学級については、学年コードに「n/a：該当なし」を設定することにより表現する。

リクエストパラメータとクラス図との対応関係を表 3-23、表 3-24 に示す。

表 3-23 学級情報 API (`getAllClassInfo`) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID
date	学級情報	存在期間

表 3-24 学級情報 API (`getClassInfo`) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID
class_id	学級情報	学級 ID
date	学級情報	存在期間

3.4.4 動的振る舞い

学級情報 API サービスのユースケース記述を表 3-25 に示す。ここでは例として、学級情報が校務系システムに登録されている前提で、それを授業・学習系システムに連携する場合について解説する。

表 3-25 学級情報 API サービスのユースケース記述

項目	内容
ユースケース名	学級情報を連携する。
概要	校務系システムに登録されている学級情報（学級名、存在期間、学年、学級担任、学級副担任）を、授業・学習系システムに連携する。
目的	システム管理業務を効率化するため。
アクター	- 授業・学習系システム - 学級情報 API サービス（校務系システム） - 学級情報（校務系システム）
開始条件	（ユースケースを開始する条件。開始のきっかけ・トリガー） 授業・学習系システムから校務系システムの学級情報 API を呼び出す。
事前条件	（ユースケースが開始される前に満たされている条件） 暫定学校 ID は校務系システムと授業・学習系システム間でマスタデータとして共有され、両システムに登録されていること。
正常フロー	（正常に処理が進んだ場合） 1. 授業・学習系システムは、学校、日付を指定して、校務系システムに学級 ID の一覧を要求する。 2. 校務系システムは、指定された学校、日付の学級 ID の一覧を抽出する。 3. 校務系システムは、授業・学習系システムに、2 の学級 ID の一覧を返す。 4. 授業・学習系システムは、学校、学級 ID を指定して、校務系システムに学級情報を要求する。 5. 校務系システムは、指定された学校、学級 ID の学級情報を取得する。 6. 校務系システムは、授業・学習系システムに 5 の学級情報を返す。 7. 授業・学習系システムは、6 の学級情報を自システムに登録する。次の学級名がある場合は、4 の処理に戻る。
代替フロー	（エラーが発生した場合。代替フローを実行しても事後条件は満たされる） 上記 2 で学級 ID 件数が 0 件だった場合（HTTP ステータスコード：200）、ユースケースを終了する。
例外フロー	（ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない） ・ 上記 5 で学級情報の取得に失敗した場合（HTTP ステータスコード：404）、ユースケースを中断する。 ・ 上記 5 で学級情報が空の場合（HTTP ステータスコード：200）、ユースケー

項目	内容
	スを中断する。
事後条件	(ユースケースが実行された後に満たされている条件) 授業・学習系システムと校務系システムに登録されている学級情報の同期がとれていること。

学級情報 API サービスのイベントフロー図を図 3-28 に示す。

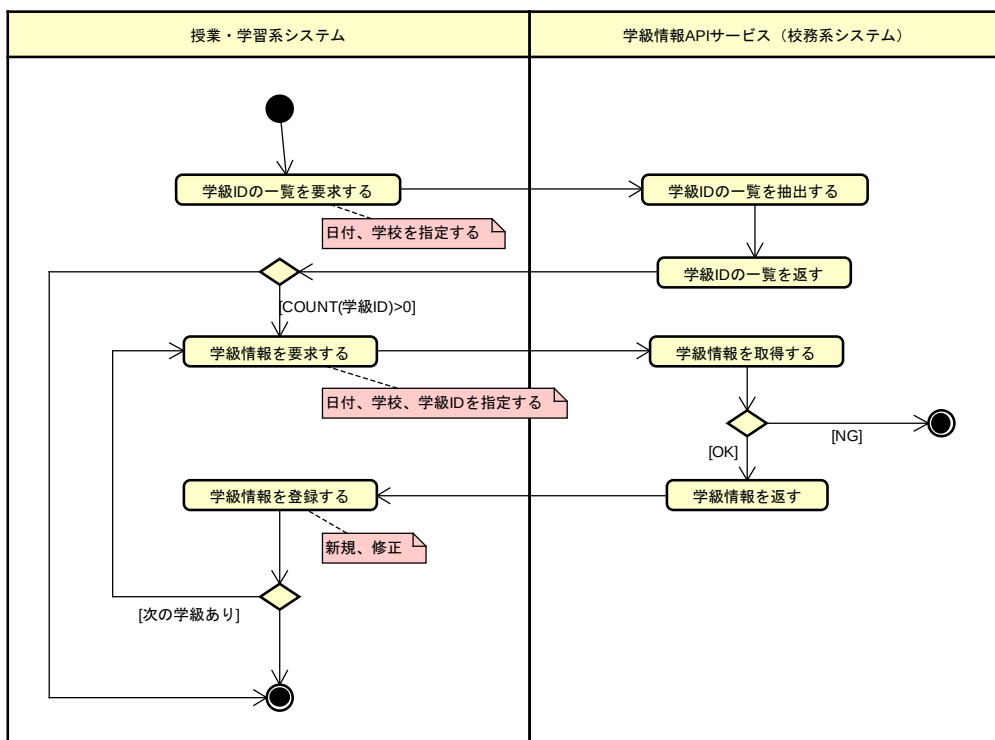


図 3-28 イベントフロー（学級情報 API サービス）

暫定学校 ID は校務系システムと授業・学習系システムとの間で事前に共有されているものとする。

学級情報 API サービスのシーケンス図を図 3-29 に示す。

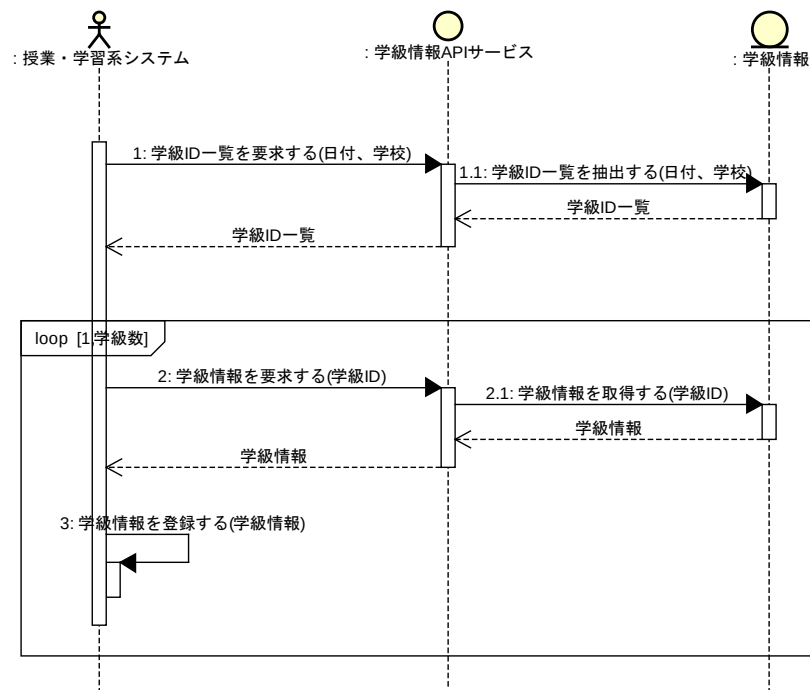


図 3-29 シーケンス図（学級情報 API サービス）

最初に API サービス `getAllClassInfo` で対象となる学級 ID の一覧を取得する（図中 1）。

取得した学級 ID 一覧から、学級 ID を指定して API サービス `getClassInfo` を繰り返し呼び出すことにより（図中 2）、当該学級情報を取得し、授業・学習系システムに登録する（図中 3）。

3.5 異動情報 API サービス

3.5.1 仕様概要

3.5.1.1 getAllTransferInfo

(1) 説明

指定した児童生徒、期間、異動種別の異動日一覧を返す。

(2) パス

GET /transfer_info/{localstudent_id}

(3) リクエストパラメータ

表 3-26 異動情報 API (getAllTransferInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	localstudent_id	○	システム連携用児童生徒共通 ID
クエリ	since	○	期間（開始日）(YYYY-MM-DD)
クエリ	until	○	期間（終了日）(YYYY-MM-DD)
クエリ	transfercode		異動コード

(4) レスポンス

表 3-27 異動情報 API (getAllTransferInfo) レスポンス

プロパティ名	多重度	説明
eventdate	0..*	異動日の配列

(5) サンプル

① リクエスト

GET /transfer_info/S0000017?since=2012-04-01&until=2018-03-31

② レスポンス

```
{  
  "results": [  

```

```

{
    "eventdate": [
        "2012-04-01",
        "2018-03-31"
    ]
}
]
}

```

3.5.1.2 getTransferInfo

(1) 説明

指定した児童生徒、異動日の異動情報を返す。

(2) パス

GET /transfer_info/{localstudent_id}/{eventdate}

(3) リクエストパラメータ

表 3-28 異動情報 API (getTransferInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	localstudent_id	○	システム連携用児童生徒共通 ID
パス	eventdate	○	異動日 (YYYY-MM-DD)

(4) レスポンス

表 3-29 異動情報 API (getTransferInfo) レスポンス

プロパティ名	多重度	説明
transfercode	1	異動コード
eventdate	1	異動日
eventgrade	0..1	異動時の学年コード
temp_school_id	0..1	異動先の暫定学校 ID
note	0..1	備考

(5) サンプル

① リクエスト

GET /transfer_info/S0000017/2012-04-01

② レスポンス

```
{
  "results": [
    {
      "transfercode": "admit_in",
      "eventdate": "2012-04-01",
      "eventgrade": "P1",
      "temp_school_id": "P_0669420221",
      "note": ""
    }
  ]
}
```

3.5.1.3 getAllSchoolTransferInfoList

(1) 説明

指定した学校、異動種別、期間の異動情報の一覧を返す。対象となる児童生徒が、転学・退学・去校状態である場合は、該当期間内の最終の異動情報を対象とする。

(2) パス

GET /school_transfer_info/{temp_school_id}

(3) リクエストパラメータ

表 3-30 異動情報 API (getAllSchoolTransferInfoList) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	temp_school_id	○	暫定学校 ID
クエリ	transfercode		異動コード
クエリ	since	○	期間（開始日）(YYYY-MM-DD)
クエリ	until	○	期間（終了日）(YYYY-MM-DD)

(4) レスポンス

表 3-31 異動情報 API (getAllSchoolTransferInfoList) レスポンス

プロパティ名	多重度	説明
localstudent_id	1	システム連携用児童生徒共通 ID
roster_id	0..1	学籍番号
grade	1	学年コード
class_id	1	学級 ID
student_code	1	出席番号
display_name	1	表示名（氏名情報）
original_name	1	外字氏名
encoding	0..1	文字エンコーディング
s13n_name	0..1	内字氏名
kana_name	1	ふりがな（平仮名）
eventdate	1	異動日
transfercode	1	異動コード

(5) サンプル

① リクエスト

GET /school_transfer_info/J_0662711456?since=2018-05-31&until=2018-05-31

② レスポンス

```
{
  "results": [
    {
      "localstudent_id": "S0149136",
      "roster_id": "R28665",
      "grade": "J1",
      "class_id": "1 年 D 組",
      "student_code": "213",
      "displayname": {
        "original_name": "原田 夏鈴",
        "encoding": "UTF-8",
        "s13n_name": "原田 夏鈴",
        "kana_name": "はらだ かりん"
      },
      "eventdate": "2018-05-31",
      "transfercode": "transfer_in"
    },
    {
      "localstudent_id": "S0268932",
      "roster_id": "R82063",
      "grade": "J2",
      "class_id": "2 年 B 組",
      "student_code": "66",
      "displayname": {
        "original_name": "高津 輝",
        "encoding": "UTF-8",
        "s13n_name": "高津 輝",
        "kana_name": "たかつ ひかる"
      },
    },
  ],
}
```

```
    "eventdate": "2018-05-31",  
    "transfercode": "move_out"  
  }  
]  
}
```

3.5.2 機能概要

異動情報 API サービスのユースケース図を図 3-30 に示す。

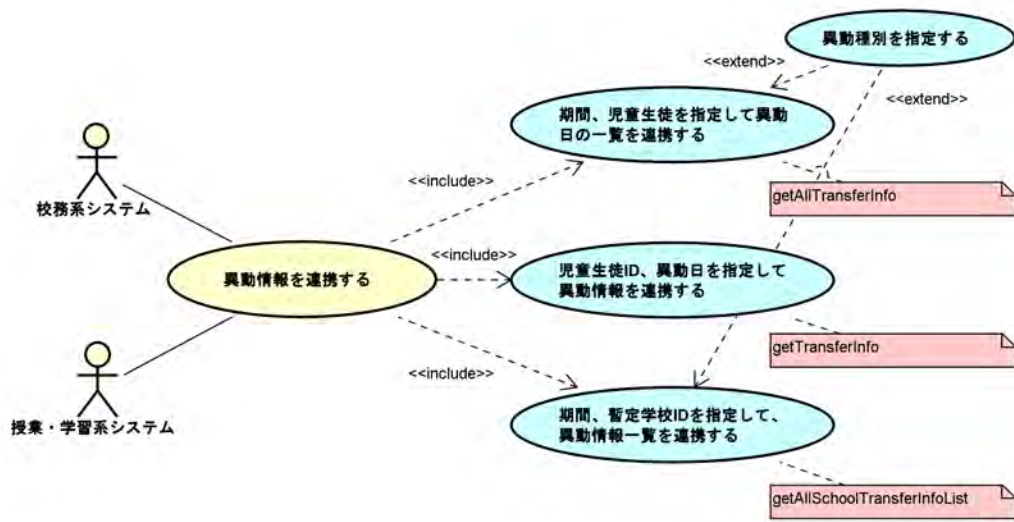


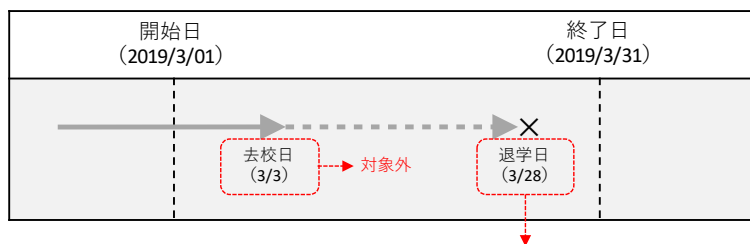
図 3-30 ユースケース図（異動情報 API サービス）

API サービス `getAllTransferInfo` は、指定した期間（since、until）、システム連携用児童生徒共通 ID（`localstudent_id`）の異動日（`eventdate`）の一覧を返す。また、期間、システム連携用児童生徒共通 ID と合わせて、異動種別を指定することにより、特定の異動種別（入学、編入学、転入学、転学、退学等、卒業、去校）に絞った異動日一覧を取得することもできる。

スマートスクール・プラットフォーム技術仕様における「システム連携用児童生徒共通 ID」は、連携するシステム間において一意である児童生徒の識別子であるため、本 API サービスにより連携できる異動情報は、連携システム間のみの異動情報に制限される点に留意する必要がある。

API サービス `getTransferInfo` では、指定したシステム連携用児童生徒共通 ID（`localstudent_id`）、異動日（`eventdate`）の異動情報を取得することができる。

また、異動情報 API サービスとして、指定した期間（since、until）、暫定学校 ID における児童生徒の直近の異動情報の一覧を返す API サービス `getAllSchoolTransferInfoList` が用意されている。`getAllSchoolTransferInfoList` では、対象となる児童生徒がすでに転学、退学、去校していた場合、指定した期間（since、until）の最終の異動情報を返す点に留意する必要がある。（図 3-31）



【出力例】

```
{
  "results": [
    {
      "localstudent_id": "S0017285",
      "grade": "J1",
      "classname": "1年E組",
      "student_code": "1",
      "displayname": {
        "original_name": "大森 都",
        "kana_name": "おおもり みやこ"
      },
      "eventdate": "2019-3-28",
      "transfercode": "withdraw"
    }
  ]
}
```

図 3-31 getAllSchoolTransferInfoList の出力例

3.5.3 静的構造

異動情報のクラス図を図 3-32 に示す。

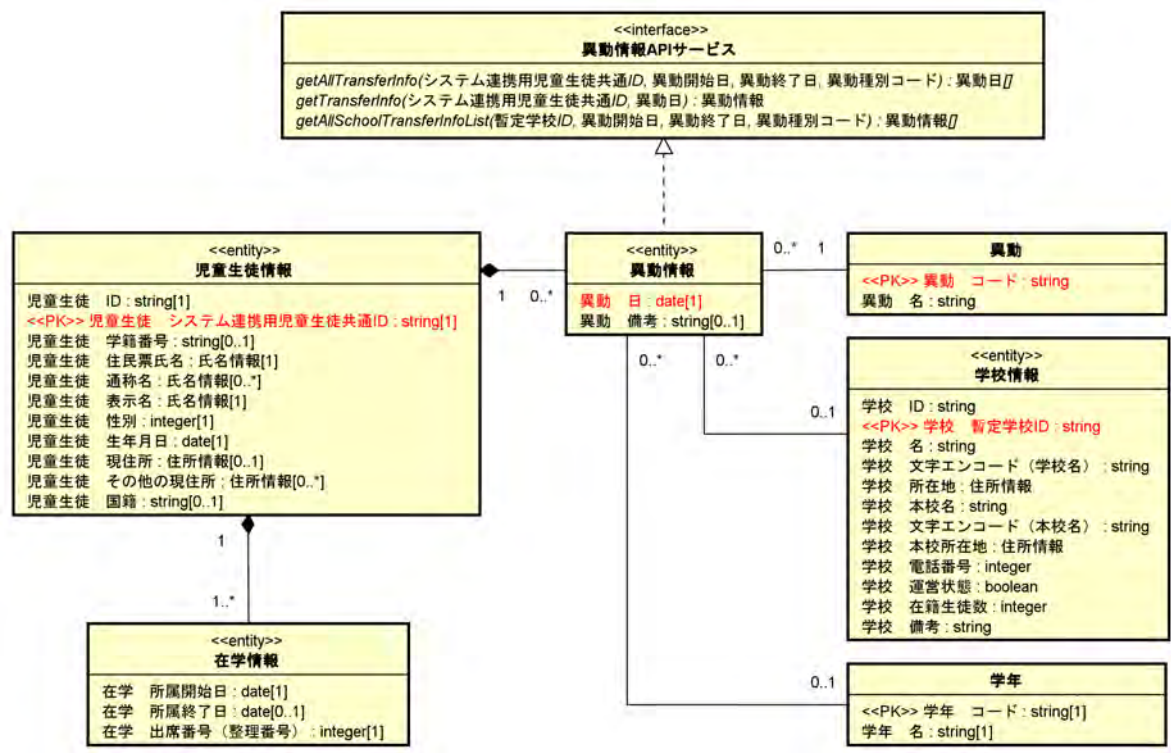


図 3-32 クラス図（異動情報 API サービス）

図 3-32 では、児童生徒の異動情報に、児童生徒情報、異動種別（入学、編入学、転入学、転学、退学等、卒業、去校）、学校情報を持つことにより、各 API サービス（getAllTransferInfo、getTransferInfo、getAllSchoolTransferInfoList）において、対象とする異動情報（一覧）を取得することができる。

異動情報には、同一校内での異動（進級、転級）は含まれない（表 3-35 参照）。
リクエストパラメータとクラス図との対応関係を表 3-32、表 3-33、表 3-34 に示す。

表 3-32 異動情報 API（getAllTransferInfo）クラス図対応表

パラメータ名	クラス名	プロパティ名
localstudent_id	児童生徒情報	システム連携用児童生徒共通 ID
since	異動情報	異動日
until	異動情報	異動日
transfercode	異動	異動コード

表 3-33 異動情報 API (getTransferInfo) クラス図対応表

パラメータ名	クラス名	プロパティ名
localstudent_id	児童生徒情報	システム連携用児童生徒共通 ID
eventdate	異動情報	異動日

表 3-34 異動情報 API (getAllSchoolTransferInfoList) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID
transfercode	異動	異動コード
since	異動情報	異動日
until	異動情報	異動日

表 3-35 在学情報、異動情報の作成タイミング（再掲）

作成タイミング	在学情報		異動情報	
入学時	○	⇒ 図 3-15	○	⇒ 図 3-33
転入学時	○		○	
編入学時	○		○	
進級時（同一校内）	○	⇒ 図 3-16	×	—
転級時（同一校内）	○		×	
転学（転出）時	○	⇒ 図 3-17	○	⇒ 図 3-34
退学等時	△	⇒ 図 3-18	○	
卒業時	△	⇒ 図 3-19	○	
去校時	×	—	○	

○…作成される △…更新される ×…作成されない

3.5.3.1 異動情報（入学、転入学、編入学時）

児童生徒が新年度初日に小学校に入学した場合の異動情報を図 3-33 に示す。転入学時、編入学時も異動種別が異なることをのぞき同様である。

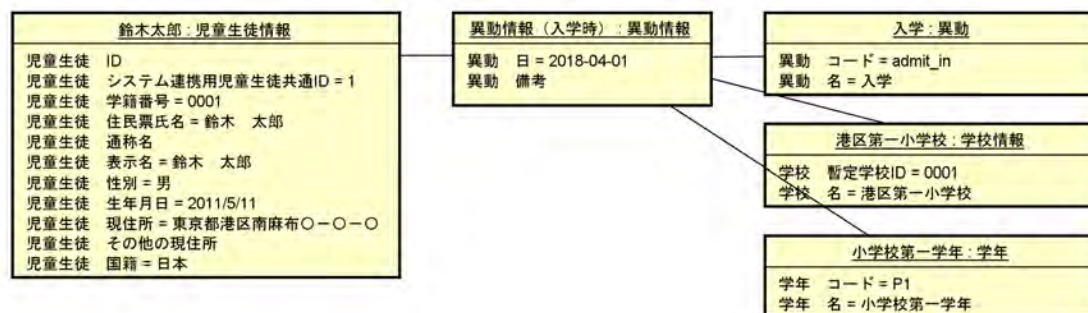


図 3-33 異動情報（入学時）

3.5.3.2 異動情報（転学、退学等、卒業、去校時）

図 3-33 で示した児童生徒が第一学年の途中（8 月 3 日）で転学（転出）した場合の異動情報を図 3-34 に示す。退学等、卒業、去校時も異動種別が異なることをのぞき同様である。

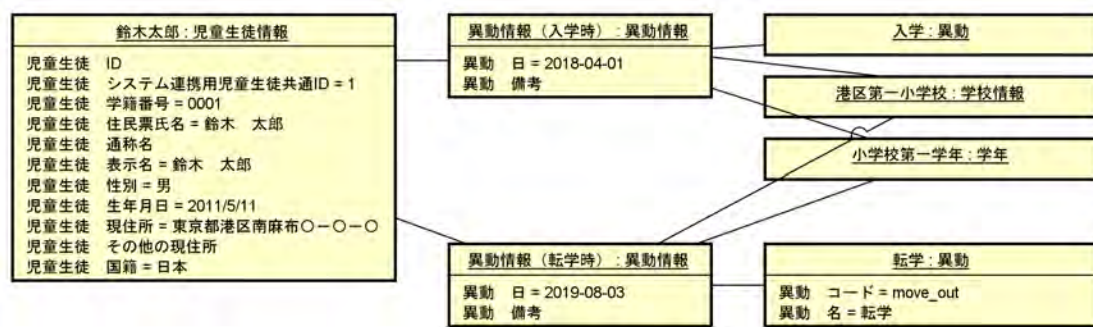


図 3-34 異動情報（転学時）

3.5.4 動的振る舞い

異動情報 API サービスのユースケース記述を表 3-36 に示す。ここでは例として、異動情報が校務系システムに登録されている前提で、それを授業・学習系システムに連携する場合について解説する。

表 3-36 異動情報 API サービスのユースケース記述

項目	内容
ユースケース名	異動情報を連携する。
概要	校務系システムに登録されている児童生徒の異動情報（入学、編入学、転入学、転学、退学等、卒業、去校）を、授業・学習系システムに連携する。
目的	システム管理業務を効率化するため。
アクター	- 授業・学習系システム - 異動情報 API サービス（校務系システム） - 異動情報（校務系システム）
開始条件	（ユースケースを開始する条件。開始のきっかけ・トリガー） 授業・学習系システムから校務系システムの異動情報 API を呼び出す。
事前条件	（ユースケースが開始される前に満たされている条件） 暫定学校 ID は校務系システムと授業・学習系システム間でマスタデータとして共有され、両システムに登録されていること。

項目	内容
	授業・学習系システムと校務系システムに登録されている児童生徒情報の同期がとれていること。（「3.2 児童生徒情報 API サービス」参照）
正常フロー	（正常に処理が進んだ場合） （正常フロー 1） - 授業・学習系システムは、期間、児童生徒、異動種別を指定して（異動種別は任意）、校務系システムに異動日の一覧を要求する。 - 校務系システムは、指定された期間、児童生徒、異動種別の異動日の一覧を抽出する。 - 校務系システムは、授業・学習系システムに、2 の異動日の一覧を返す。 - 授業・学習系システムは、システム連携用児童生徒共通 ID、異動日を指定して、校務系システムに異動情報を要求する。 - 校務系システムは、指定されたシステム連携用児童生徒共通 ID、異動日の異動情報を取得する。 - 校務系システムは、授業・学習系システムに 5 の異動情報を返す。 - 授業・学習系システムは、6 で受け取った異動情報を自システムに登録する。次の異動日がある場合は、4 の処理に戻る。 （正常フロー 2） - 授業・学習系システムは、期間、学校、異動種別を指定して（異動種別は任意）、校務系システムに児童生徒の異動情報を要求する。 - 校務系システムは、指定された期間、学校、異動種別の異動情報を取得する。 - 校務系システムは、授業・学習系システムに、2 の異動情報を返す。 - 授業・学習系システムは、3 で受け取った異動情報を自システムに登録する。
代替フロー	（エラーが発生した場合。代替フローを実行しても事後条件は満たされる） （代替フロー 1） 上記正常フロー 1 の 2 で異動日件数が 0 件だった場合（HTTP ステータスコード：200）、ユースケースを終了する。 （代替フロー 2） 上記正常フロー 2 の 2 で異動情報が 0 件だった場合（HTTP ステータスコード：200）、ユースケースを終了する。
例外フロー	（ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない） （例外フロー 1） 上記正常フロー 1 の 5 で異動情報の取得に失敗した場合（HTTP ステータスコード：404）、ユースケースを中断する。 （例外フロー 2） 上記正常フロー 2 の 2 で異動情報の取得に失敗した場合（HTTP ステータスコード：404）、ユースケースを中断する。
事後条件	（ユースケースが実行された後に満たされている条件） 授業・学習系システムと校務系システムに登録されている児童生徒の異動情報の同期がとれていること。

異動情報 API サービスのイベントフロー図を図 3-35、図 3-36 に示す。

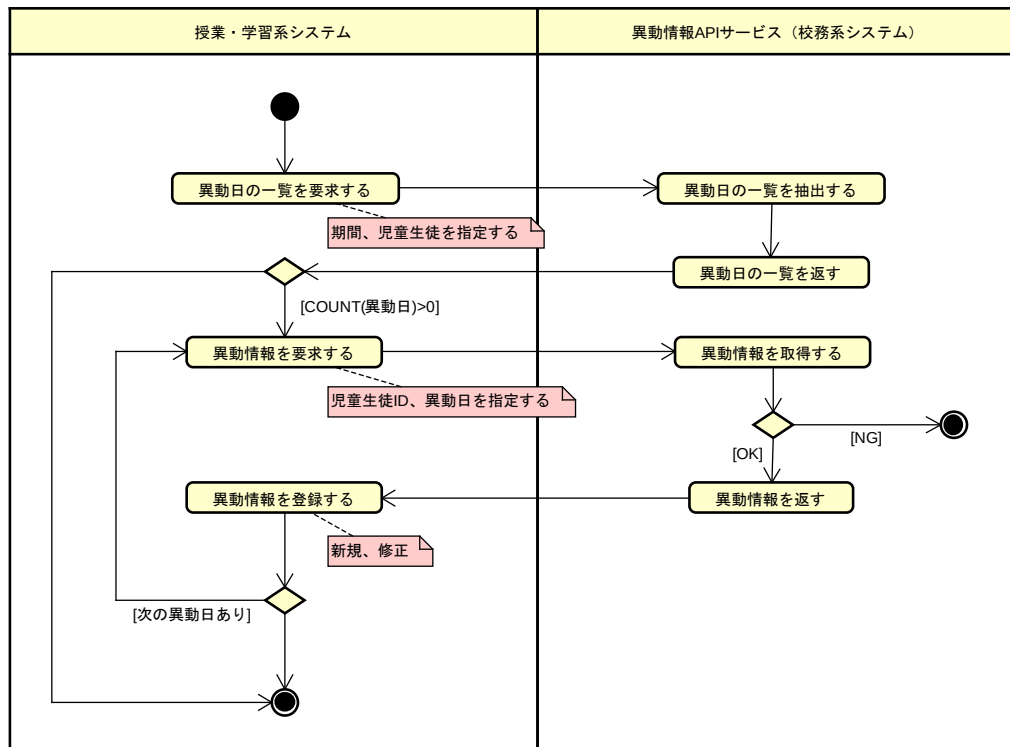


図 3-35 イベントフロー図（異動情報 API サービス）（期間、児童生徒を指定）

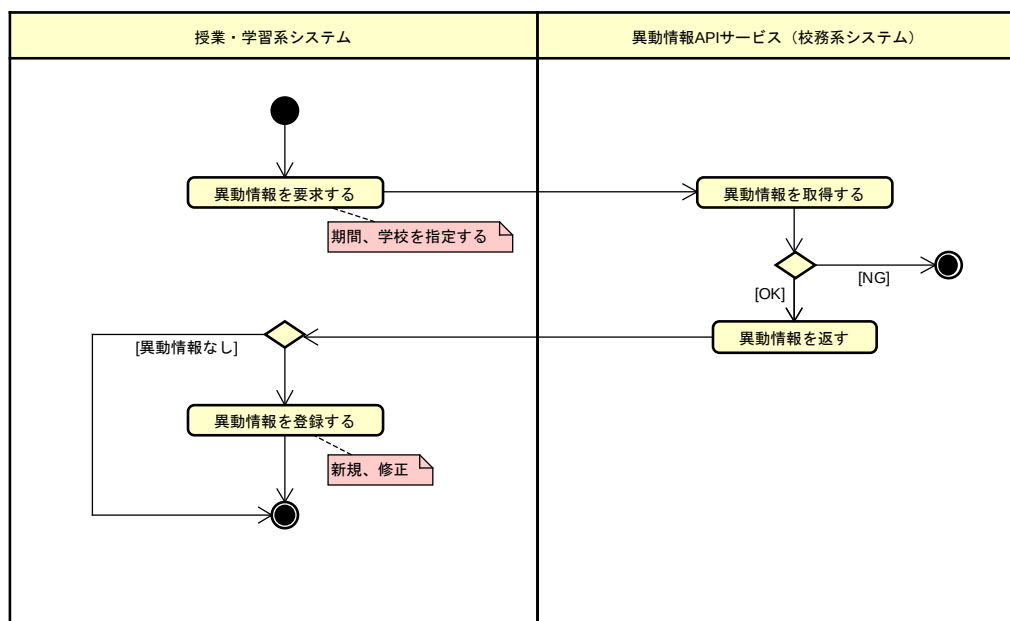


図 3-36 イベントフロー図（異動情報 API サービス）（期間、学校を指定）

暫定学校 ID は校務系システムと授業・学習系システムとの間で事前に共有されているものとする。また、システム連携用児童生徒共通 ID についても、児童生徒情報 API サービスによるデータ連携により、校務系システムと授業・学習系システムとの間で事前に共有されているものとする。

異動情報 API サービスのシーケンス図を図 3-37 に示す。

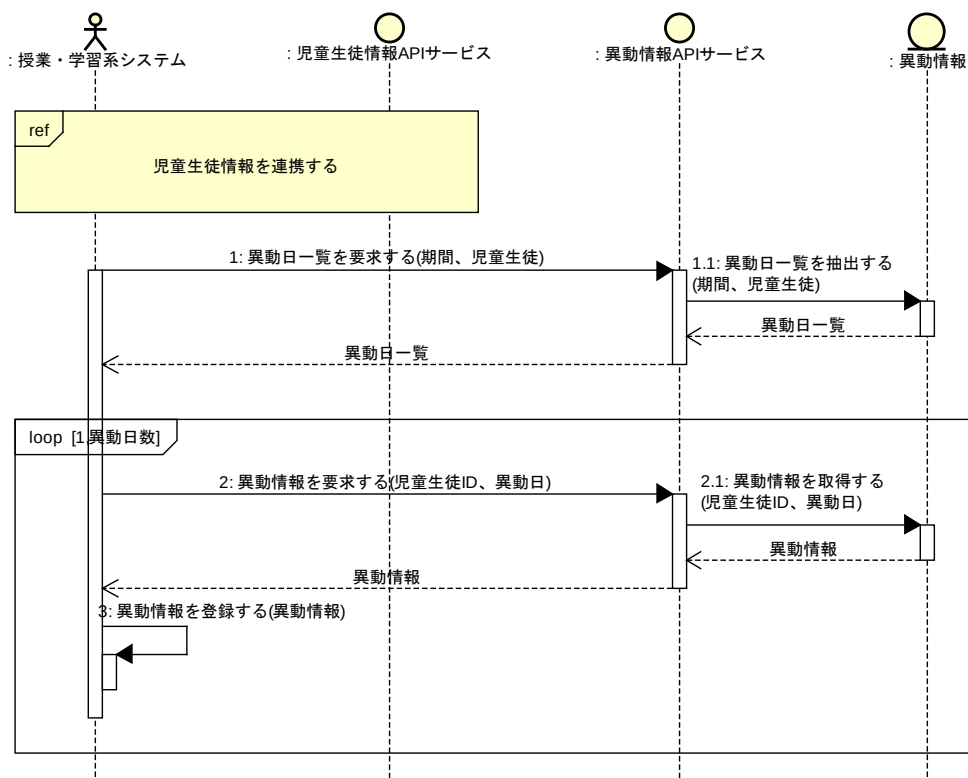


図 3-37 シーケンス図（異動情報 API サービス）（期間、児童生徒を指定する）

児童生徒個人の異動情報を取得する場合、最初に API サービス `getAllTransferInfo` で対象となるシステム連携用児童生徒共通 ID の異動日一覧を取得する（図中 1）。システム連携用児童生徒共通 ID と、取得した異動日一覧からの異動日を指定して API サービス `getTransferInfo` を繰り返し呼び出すことにより（図中 2）、当該児童生徒の異動情報を取得し、授業・学習系システムに登録する（図中 3）。

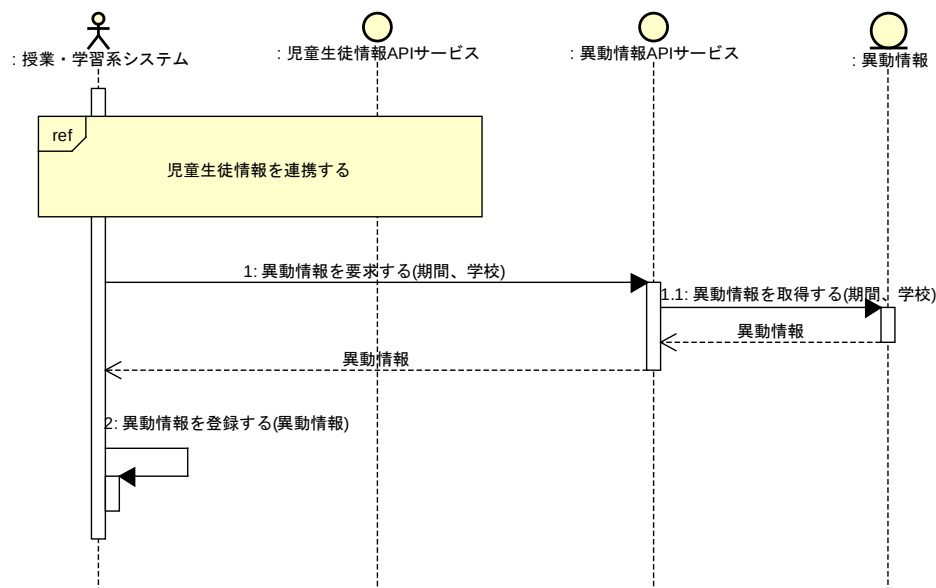


図 3-38 シーケンス図（異動情報 API サービス）（期間、学校を指定する）

学校に所属する児童生徒全員の異動情報一覧を取得する場合、暫定学校 ID、期間を指定して API サービス `getAllSchoolTransferInfoList` を呼び出すことにより（図中 1）、当該児童生徒全員の異動情報一覧を取得し、授業・学習システムに登録する（図中 2）。

3.6 出欠情報 API サービス

3.6.1 仕様概要

3.6.1.1 getAttendanceRecordInfo

(1) 説明

指定した学校、児童生徒、期間の出欠情報の一覧を返す。本項目のレスポンス値は、jCal の仕様に準拠する。

(2) パス

GET /attendance_record_info/{temp_school_id}/{localstudent_id}

(3) リクエストパラメータ

表 3-37 出欠情報 API (getAttendanceRecordInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	temp_school_id	○	暫定学校 ID
パス	localstudent_id	○	システム連携用児童生徒共通 ID
クエリ	since	○	期間（開始日）(YYYY-MM-DD)
クエリ	until	○	期間（終了日）(YYYY-MM-DD)

(4) レスポンス

表 3-38 出欠情報 API (getAttendanceRecordInfo) レスポンス

プロパティ名	多重度	説明
vcalendar	1	jCal ルート要素
calscale	1	採用暦 (" GREGORIAN" 固定)
prodid	1	製品 ID
version	1	jCal バージョン ("2.0" 固定)
vevent	1..*	jCal の本文要素単位
dtstart	1	出欠日 (YYYY-MM-DD)
dtend	1	出欠日の翌日 (YYYY-MM-DD)
categories	1	出欠理由コード (複数設定可)
description	1	備考

(5) サンプル

① リクエスト

```
GET /attendance_record_info/J_0662711456/S00000059?since=2018-04-10&until=2018-04-10
```

② レスポンス

```
{
  "results": [
    [
      "vcalendar",
      [
        [
          "calscale",
          {},
          "text",
          "GREGORIAN"
        ],
        [
          "prodid",
          {},
          "text",
          "https://smartschool.example.org/sspfapi/v1/"
        ],
        [
          "version",
          {},
          "text",
          "2.0"
        ]
      ]
    ]
  ]
}
```



```

],
[
  [
    "vevent",
    [
      [
        "dtstart",
        {},
        "date",
        "2018-04-10"
      ],
      [
        "dtend",
        {},
        "date",
        "2018-04-11"
      ],
      [
        "categories",
        {},
        "text",
        "atnd"
      ],
      [
        "description",
        {},
        "text",
        ""
      ]
    ]
  ]
]

```

```
    ]  
  ]  
}
```

3.6.2 機能概要

出欠情報 API サービスのユースケース図を図 3-39 に示す。

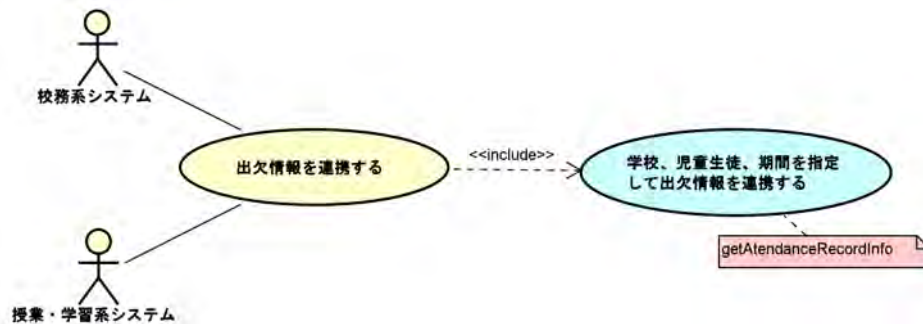


図 3-39 ユースケース図（出欠情報 API サービス）

API サービス `getAttendanceRecordInfo` は、指定した暫定学校 ID (`temp_school_id`)、システム連携用児童生徒共通 ID、期間 (`since`、`until`) の出欠情報を返す。

スマートスクール・プラットフォーム技術仕様における「出欠情報」は、カレンダー情報の規定に関する国際標準仕様である jCal の `vevent` の配列として表現する。jCal の仕様上、時刻情報も含めた出欠情報の表現も可能であるが、スマートスクール・プラットフォーム技術仕様における出欠情報は日単位での出欠情報とする。

連携する出欠情報としては、学習指導要録上の出欠の記録（出席、欠席、病欠・事故欠、出席停止・忌引き等、遅刻、早退）に加えて、スマートスクール・プラットフォーム技術仕様では、学習指導、生活指導の観点から連携情報として必要と考えられる出席認定、転入前、去校後、転出後、休日、不明が追加されている。

【出欠理由】

- 学習指導要録上の出欠の記録
出席、病欠・事故欠、出席停止・忌引き等、遅刻、早退
- ・ スマートスクール・プラットフォーム技術仕様にて追加した出欠理由
出席認定、転入前、去校後、転出後、休日、不明

API サービス `getAttendanceRecordInfo` の返り値について、指定した期間(`since`、`until`)内に 1 日以上、当該学校に在籍していた児童生徒が対象となる。当該児童生徒の出欠情

報がない場合は、当該日付の出欠理由に「不明」を設定してもよい。

3.6.3 静的構造

出欠情報のクラス図を図 3-40 に示す。

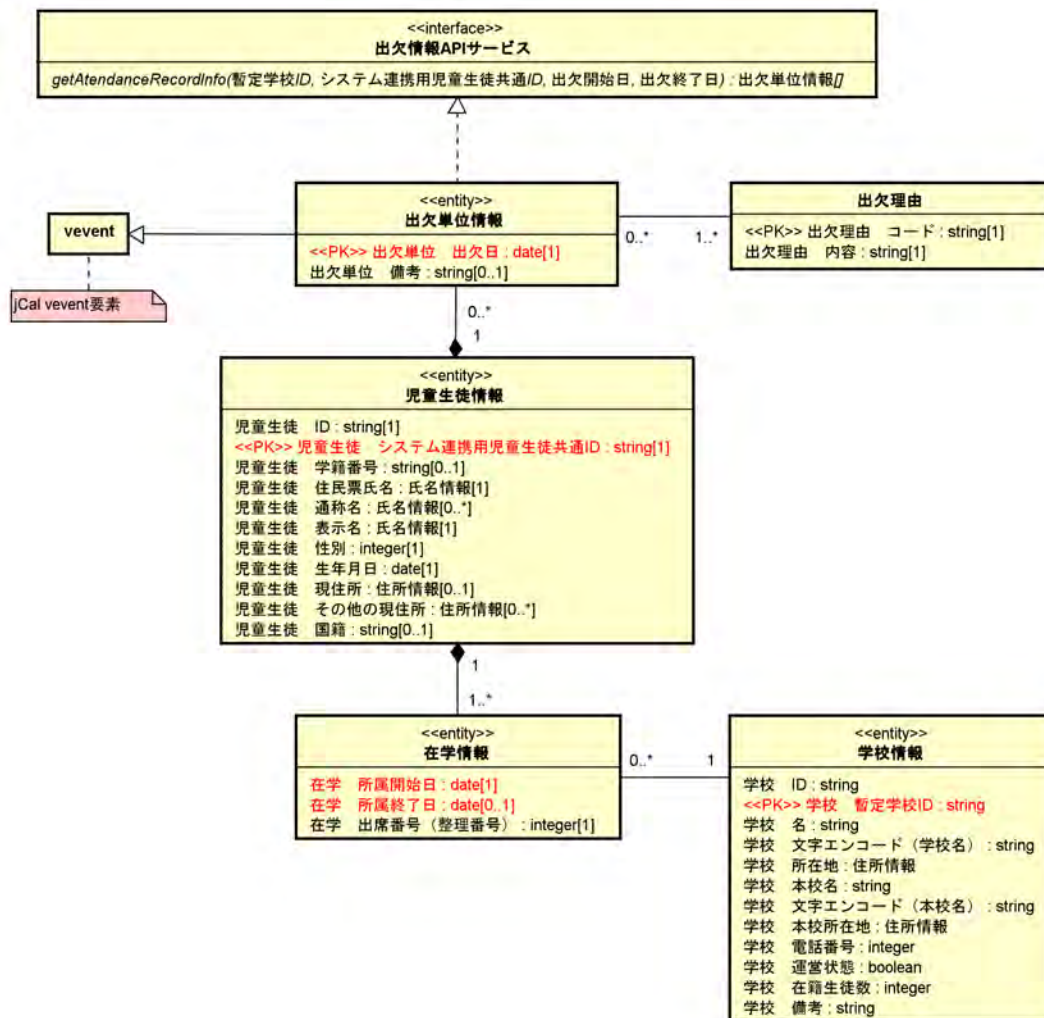


図 3-40 クラス図（出欠情報 API サービス）

スマートスクール・プラットフォーム技術仕様では、児童生徒の出欠情報を jCal の vevent である出欠単位情報の配列で表現する。

出欠単位情報には、児童生徒情報、在学情報および学校情報が関連付けられていることにより、API サービス getAttendanceRecordInfo で、対象の出欠情報を取得する。その際に対象となる在学情報は出欠日が所属期間（所属開始日、所属終了日）内にあるものが対象となることに注意する必要がある。

リクエストパラメータとクラス図との対応関係を表 3-39 に示す。

表 3-39 出欠情報 API (getAttendanceInfo) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID
localstudent_id	児童生徒情報	システム連携用児童生徒共通 ID
since	在学情報	所属開始日／所属終了日
until	在学情報	所属開始日／所属終了日

3.6.4 動的振る舞い

出欠情報 API サービスのユースケース記述を表 3-40 に示す。ここでは例として、出欠情報が校務系システムに登録されている前提で、それを授業・学習系システムに連携する場合について解説する。

表 3-40 出欠情報 API サービスのユースケース記述

項目	内容
ユースケース名	出欠情報を連携する。
概要	校務系システムに登録されている児童生徒の出欠情報（出席、病欠・事故欠、出席停止・忌引き等、遅刻、早退、出席認定、転入前、去校後、転出後、休日、不明）を、授業・学習系システムに連携する。
目的	（対応する文科省データ活用パターン） 学習指導の充実 - 一人ひとりに応じた学習指導 生徒指導の充実 - データに基づいた生活指導（担当教諭） - 児童生徒に対する組織的支援（養護教諭、スクールカウンセラー、スクールソーシャルワーカー） 学校経営の充実 - 管理職による児童生徒の把握（学校管理職）
アクター	- 授業・学習系システム - 出欠情報 API サービス（校務系システム） - 出欠情報（校務系システム）
開始条件	（ユースケースを開始する条件。開始のきっかけ・トリガー） 授業・学習系システムから校務系システムの出欠情報 API を呼び出す。
事前条件	（ユースケースが開始される前に満たされている条件） - 暫定学校 ID は校務系システムと授業・学習系システム間でマスターデータとして共有され、両システムに登録されていること。 - 授業・学習系システムと校務系システムに登録されている児童生徒情報の同期がとれていること。（「3.2 児童生徒情報 API サービス」参照）
正常フロー	（正常に処理が進んだ場合） - 授業・学習系システムは、学校、児童生徒、期間を指定して、校務系システムに児童生徒の出欠情報を要求する。 - 校務系システムは、指定された学校、児童生徒、期間の出欠情報を取得する。 - 校務系システムは、授業・学習系システムに、2 の出欠情報を返す。 - 授業・学習系システムは、3 で受け取った出欠情報を自システムに登録する。
代替フロー	（エラーが発生した場合。代替フローを実行しても事後条件は満たされる）

項目	内容
	—
例外フロー	(ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない) 上記 2 で出欠情報の取得に失敗した場合 (HTTP ステータスコード: 404)、ユースケースを中断する。
事後条件	(ユースケースが実行された後に満たされている条件) 授業・学習系システムと校務系システムに登録されている出欠情報の同期がとれていること。

出欠情報 API サービスのイベントフロー図を図 3-41 に示す。

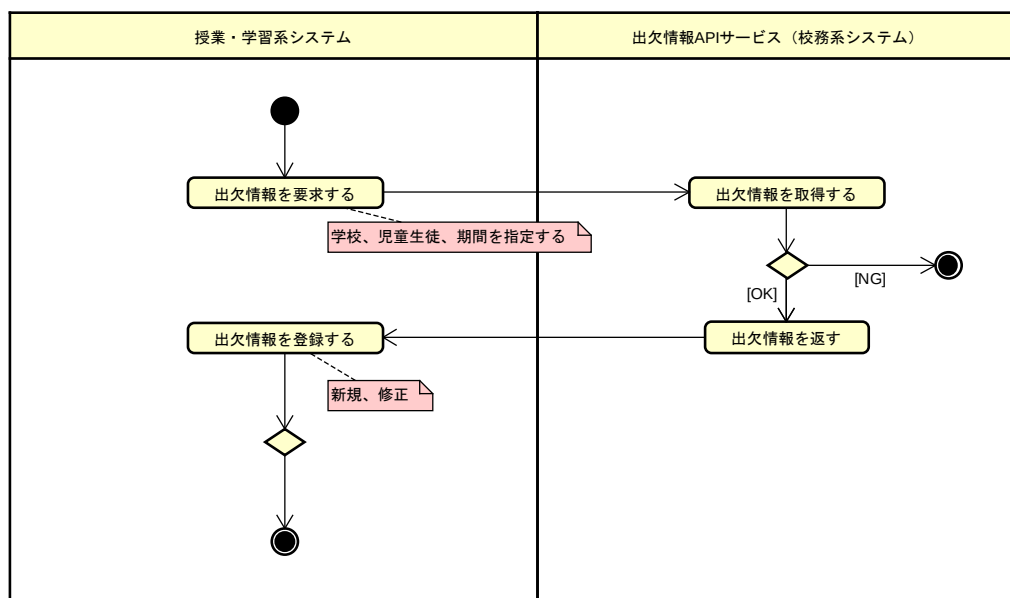


図 3-41 イベントフロー図（出欠情報 API サービス）

暫定学校 ID は校務系システムと授業・学習系システムとの間で事前に共有されているものとする。また、システム連携用児童生徒共通 ID についても、児童生徒情報 API サービスによるデータ連携により、校務系システムと授業・学習系システムとの間で事前に共有されているものとする。

出欠情報 API サービスのシーケンス図を図 3-42 に示す。

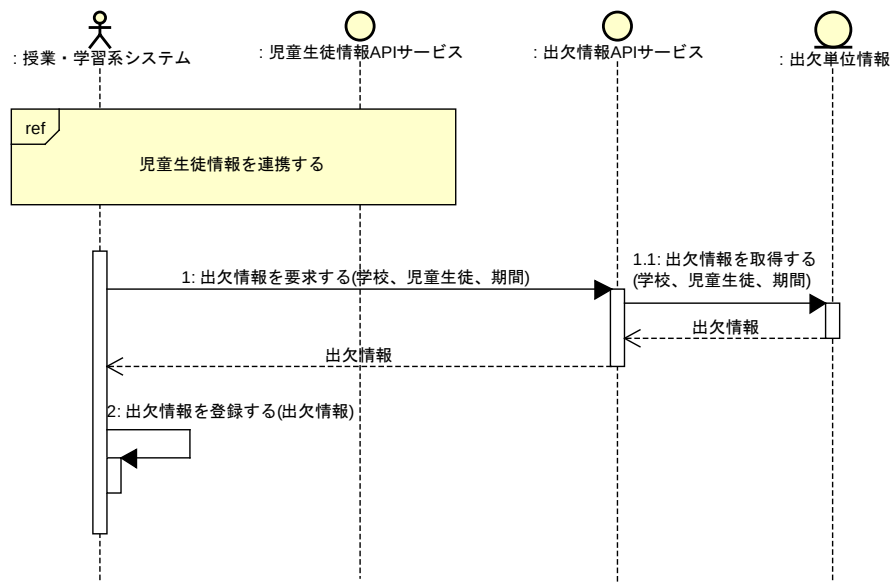


図 3-42 シーケンス図（出欠情報 API サービス）

児童生徒の出欠情報（出欠単位情報一覧）を取得するには、暫定学校 ID、システム連携用児童生徒共通 ID、期間を指定して API サービス `getAttendanceRecordInfo` を呼び出すことにより（図中 1）、当該児童生徒の出欠情報を取得し、授業・学習系システムに登録する（図中 2）。

3.7 時間割情報 API サービス

3.7.1 仕様概要

3.7.1.1 getTimetableInfo

(1) 説明

指定した学校、学級、期間の取得時点における時間割情報の一覧を返す。本項目のレスポンス値は、jCal の仕様および以下の規定に準拠する。

(2) パス

GET /timetable_info/{temp_school_id}

(3) リクエストパラメータ

表 3-41 時間割情報 API (getTimetableInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	temp_school_id	○	暫定学校 ID
クエリ	class_id	○	学級 ID
クエリ	since	○	期間（開始日）(YYYY-MM-DD)
クエリ	until	○	期間（終了日）(YYYY-MM-DD)

(4) レスポンス

表 3-42 時間割情報 API (getTimetableInfo) レスポンス

プロパティ名	多重度	説明
vcalendar	1	jCal ルート要素
calscale	1	採用暦 (" GREGORIAN" 固定)
prodid	1	製品 ID
version	1	jCal バージョン ("2.0" 固定)
vevent	1..*	jCal の本文要素単位
dtstart	1	日付 (YYYY-MM-DD)
dtend	1	翌日日付 (YYYY-MM-DD)
categories	1	コマ内容。学校場面コードと教科・科目コードを用いることができる。1~100%の指定により分割授業を表現することも可能。

(5) サンプル

① リクエスト

GET

/timetable_info/J_0662711456?class_id=1%E5%B9%B4A%E7%B5%84&since=2018-04-06&until=2018-04-06

② レスポンス

```
{
  "results": [
    [
      "vcalendar",
      [
        [
          "calscale",
          {},
          "text",
          "GREGORIAN"
        ],
        [
          "prodid",
          {},
          "text",
          "https://api.smartschool-platform.jp/v1/"
        ],
        [
          "version",
          {},
          "text",
          "2.0"
        ]
      ]
    ]
  ]
}
```

```

    ]
  ],
  [
    [
      "vevent",
      [
        [
          "dtstart",
          {},
          "date",
          "2018-04-06"
        ],
        [
          "dtend",
          {},
          "date",
          "2018-04-07"
        ],
        [
          "categories",
          {},
          "text",
          "1st_period",
          "100%",
          "J080_01Y"
        ]
      ]
    ]
  ],
  [
    "vevent",
    [

```

```

[
  "dtstart",
  {},
  "date",
  "2018-04-06"
],
[
  "dtend",
  {},
  "date",
  "2018-04-07"
],
[
  "categories",
  {},
  "text",
  "2nd_period",
  "100%",
  "J510Y"
]
],
[
  "vevent",
  [
    [
      "dtstart",
      {},
      "date",
      "2018-04-06"
    ],

```

```

[
  "dtend",
  {},
  "date",
  "2018-04-07"
],
[
  "categories",
  {},
  "text",
  "3rd_period",
  "100%",
  "J010"
]
],
[
  "vevent",
  [
    [
      "dtstart",
      {},
      "date",
      "2018-04-06"
    ],
    [
      "dtend",
      {},
      "date",
      "2018-04-07"
    ]
  ],

```

```

[
    "categories",
    {},
    "text",
    "4th_period",
    "100%",
    "J080_02Y"
]
],
[
    "vevent",
    [
        [
            "dtstart",
            {},
            "date",
            "2018-04-06"
        ],
        [
            "dtend",
            {},
            "date",
            "2018-04-07"
        ],
        [
            "categories",
            {},
            "text",
            "5th_period",
            "100%",

```

```

        "J510Y"
    ]
]
],
[
    "vevent",
    [
        [
            "dtstart",
            {},
            "date",
            "2018-04-06"
        ],
        [
            "dtend",
            {},
            "date",
            "2018-04-07"
        ],
        [
            "categories",
            {},
            "text",
            "6th_period",
            "60%",
            "J510Y_04Y"
        ]
    ]
]
],
[
    "vevent",

```



```

[
  [
    "dtstart",
    {},
    "date",
    "2018-04-06"
  ],
  [
    "dtend",
    {},
    "date",
    "2018-04-07"
  ],
  [
    "categories",
    {},
    "text",
    "6th_period",
    "40%",
    "J040"
  ]
]
]
]
]
]
]
}

```

3.7.2 機能概要

時間割情報 API サービスのユースケース図を図 3-43 に示す。

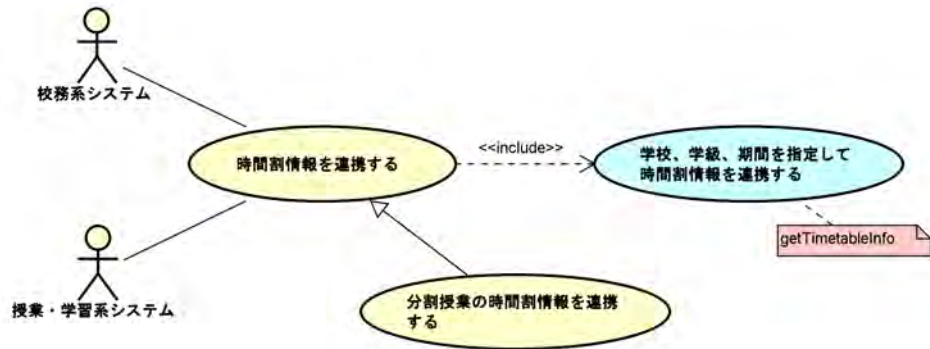


図 3-43 ユースケース図（時間割情報 API サービス）

API サービス getTimetableInfo は、指定した暫定学校 ID (temp_school_id)、学級 ID (class_id) の時間割情報を返す。

スマートスクール・プラットフォーム技術仕様における「時間割情報」は、カレンダー情報の規定に関する国際標準仕様である jCal の vevent の配列として表現する。

時間割の各コマが 1 つの vevent に対応する。また、モジュール授業などの分割授業を想定して、時間割のコマを複数に分割し、それぞれを vevent として表現することもできる。また、取得される時間割情報は各学級において問い合わせ時点での時間割情報であることに注意する必要がある。

3.7.3 静的構造

時間割情報のクラス図を図 3-44 に示す。

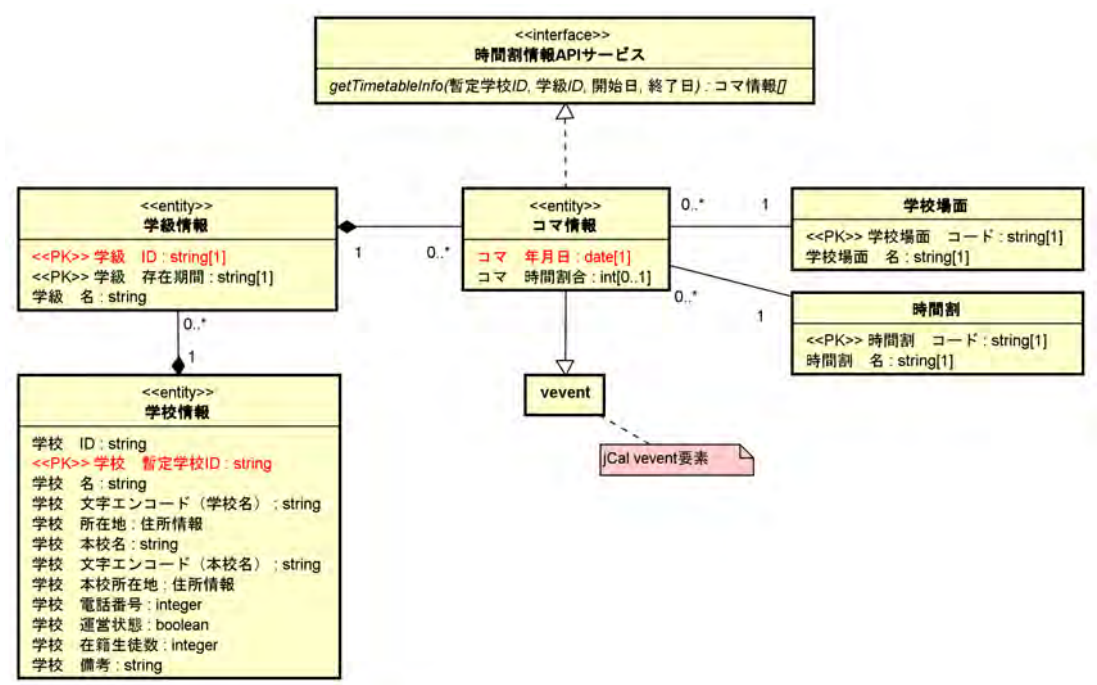


図 3-44 クラス図（時間割情報 API サービス）

スマートスクール・プラットフォーム技術仕様では、時間割情報を jCal の vevent であるコマ情報の配列で表現する。

コマ情報には、学級情報および学校情報が関連付けられていることにより、API サービス getTimetableInfo で、対象の時間割情報を取得することができる。

リクエストパラメータとクラス図との対応関係を表 3-43 に示す。

表 3-43 時間割情報 API (getTimetableInfo) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID
class_id	学級情報	学級 ID
since	コマ情報	年月日
until	コマ情報	年月日

3.7.4 動的振る舞い

時間割情報 API サービスのユースケース記述を表 3-44 に示す。ここでは例として、時間割情報が校務系システムに登録されている前提で、それを授業・学習系システムに連携する場合について解説する。

表 3-44 時間割情報 API サービスのユースケース記述

項目	内容
ユースケース名	時間割情報を連携する。
概要	校務系システムに登録されている問い合わせ時点の時間割情報（分割授業も含む）を、授業・学習系システムに連携する。
目的	（対応する文科省データ活用パターン） 学習指導の充実 一人ひとりに応じた学習指導
アクター	- 授業・学習系システム - 時間割情報 API サービス（校務系システム） - 時間割情報（校務系システム）
開始条件	（ユースケースを開始する条件。開始のきっかけ・トリガー） 授業・学習系システムから校務系システムの時間割情報 API を呼び出す。
事前条件	（ユースケースが開始される前に満たされている条件） - 暫定学校 ID は校務系システムと授業・学習系システム間でマスターデータとして共有され、両システムに登録されていること。 - 授業・学習系システムと校務系システムに登録されている学級情報の同期がとれていること。（「3.4 学級情報 API サービス」参照）
正常フロー	（正常に処理が進んだ場合） - 授業・学習系システムは、学校、学級を指定して、校務系システムに時間割情報を要求する。 - 校務系システムは、指定された学校、学級の時間割情報を取得する。 - 校務系システムは、授業・学習系システムに、2 の時間割情報を返す。 - 授業・学習系システムは、3 で受け取った時間割情報を自システムに登録する。
代替フロー	（エラーが発生した場合。代替フローを実行しても事後条件は満たされる） 上記 2 で時間割情報が 0 件の場合（HTTP ステータスコード：200）、ユースケースを終了する。
例外フロー	（ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない） 上記 2 で時間割情報の取得に失敗した場合（HTTP ステータスコード：404）、ユースケースを中断する。

項目	内容
事後条件	(ユースケースが実行された後に満たされている条件) 授業・学習系システムと校務系システムに登録されている時間割情報の同期がとれていること。

時間割情報 API サービスのイベントフロー図を図 3-45 に示す。

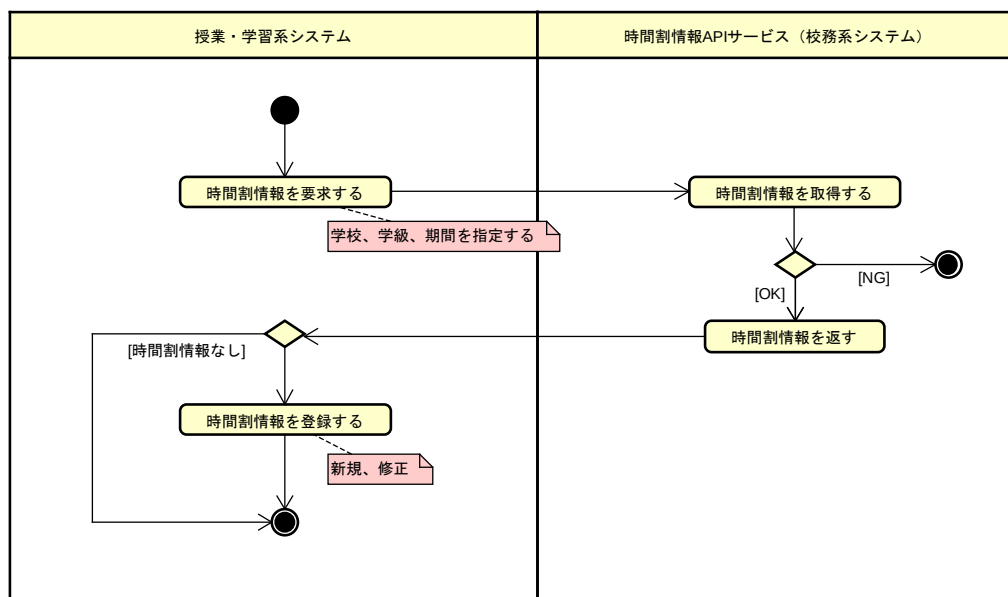


図 3-45 イベントフロー図（時間割情報 API サービス）

暫定学校 ID は校務系システムと授業・学習系システムとの間で事前に共有されているものとする。また、学級 ID についても、学級情報 API サービスによるデータ連携により、校務系システムと授業・学習系システムとの間で事前に共有されているものとする。

時間割情報 API サービスのシーケンス図を図 3-46 に示す。

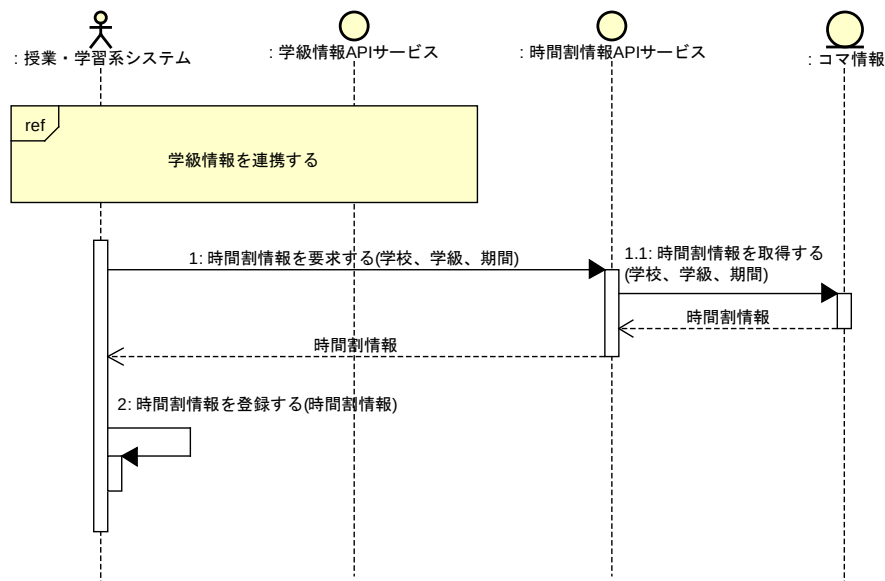


図 3-46 シーケンス図（時間割情報 API サービス）

学級の時間割情報を取得するには、暫定学校 ID、学級 ID、期間を指定して API サービス `getTimetableInfo` を呼び出すことにより（図中 1）、当該学級の時間割情報を取得し、授業・学習系システムに登録する（図中 2）。

3.8 日常所見情報 API サービス

3.8.1 仕様概要

3.8.1.1 getOpinionInfo

(1) 説明

指定した学校、児童生徒、期間の日常所見情報の一覧を返す。

(2) パス

GET /opinion_info/{temp_school_id}/{localstudent_id}

(3) リクエストパラメータ

表 3-45 日常所見情報 API (getOpinionInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	temp_school_id	○	暫定学校 ID
パス	localstudent_id		システム連携用児童生徒共通 ID
クエリ	since	○	期間（開始日）(YYYY-MM-DD)
クエリ	until	○	期間（終了日）(YYYY-MM-DD)

(4) レスポンス

表 3-46 日常所見情報 API (getOpinionInfo) レスポンス

プロパティ名	多重度	説明
localstudent_id	1	システム連携用児童生徒共通 ID
creation_date	1	作成日 (YYYY-MM-DD)
opinion_code	1..*	所見分類コード
comment	1	所見内容
localstaff_id	1	記入教職員 ID

(5) サンプル

① リクエスト

GET /opinion_info/J_0662711456/S0000059?since=2018-04-01&until=2018-05-

② レスポンス

```

{
  "results": [
    {
      "localstudent_id": "S0000059",
      "creation_date": "2018-04-26",
      "opinion_code": [
        "SS-OP004"
      ],
      "comment": "体験学習では、宿泊班のリーダーとなり、班の皆に連絡を  
正確に伝え、安全に過ごせるよう気を配ることができました。",
      "localstaff_id": "T0005514"
    },
    {
      "localstudent_id": "S0000059",
      "creation_date": "2018-05-04",
      "opinion_code": [
        "SS-OP002"
      ],
      "comment": "音楽の「〇〇〇〇」では、人一倍努力を重ね、一人で演奏  
することができるようになりました。",
      "localstaff_id": "T0005514"
    },
    {
      "localstudent_id": "S0000059",
      "creation_date": "2018-05-19",
      "opinion_code": [
        "SS-OP005"
      ]
    }
  ]
}

```



```

    ],
    "comment": "国際理解教室では、スリランカの文化に興味をもち、日本
とは異なる生活や考え方があることを理解していました。",
    "localstaff_id": "T0005514"
  },
  {
    "localstudent_id": "S0000059",
    "creation_date": "2018-05-30",
    "opinion_code": [
      "SS-OP005"
    ],
    "comment": "インターネットや本で調べた内容をわかりやすくまとめ、
発表することができました。",
    "localstaff_id": "T0005514"
  }
]
}

```

3.8.1.2 getAllOpinionCodeInfo

(1) 説明

指定した学校における所見分類コードの一覧を返す。

(2) パス

GET /opinion_info/opinion_code_info/{temp_school_id}

(3) リクエストパラメータ

表 3-47 日常所見情報 API (getOpinionInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	temp_school_id	○	暫定学校 ID

(4) レスポンス

表 3-48 日常所見情報 API (getOpinionInfo) レスポンス

プロパティ名	多重度	説明
opinion_code	1	所見分類コード
opinion_display_name	1	所見分類コード表示名
opinion_description	0..1	説明
is_deprecated	1	有効状態 (true : 有効、false : 無効)
last_modified_date	1	更新日時 (YYYY-MM-DDThh:mm:ss+09:00)

(5) サンプル

① リクエスト

GET /opinion_info/opinion_code_info/J_0662711456

② レスポンス

```
{
  "results": [
    {
      "opinion_code": "SS-OP001",
      "opinion_display_name": "学習",
      "opinion_description": "各教科や総合的な学習の時間の学習に関する  
所見",
      "is_deprecated": true,
      "last_modified_date": "2019-04-24T12:01:03+09:00"
    },
    {
      "opinion_code": "SS-OP002",
      "opinion_display_name": "行動",
      "opinion_description": "行動に関する所見",
      "is_deprecated": true,

```

```

      "last_modified_date": "2019-04-24T07:03:59+09:00"
    },
    {
      "opinion_code": "SS-OP003",
      "opinion_display_name": "特別活動",
      "opinion_description": "特別活動に関する事実及び所見",
      "is_deprecated": true,
      "last_modified_date": "2019-04-24T08:59:31+09:00"
    },
    {
      "opinion_code": "SS-OP004",
      "opinion_display_name": "総合",
      "opinion_description": "児童の成長の状況にかかわる総合的な所見",
      "is_deprecated": true,
      "last_modified_date": "2019-04-24T21:04:55+09:00"
    },
    {
      "opinion_code": "SS-OP005",
      "opinion_display_name": "外国語",
      "opinion_description": "外国語授業に関する所見",
      "is_deprecated": true,
      "last_modified_date": "2019-04-24T13:01:53+09:00"
    },
    {
      "opinion_code": "SS-OP006",
      "opinion_display_name": "道徳",
      "opinion_description": "道徳授業に関する所見",
      "is_deprecated": true,
      "last_modified_date": "2019-04-24T19:05:59+09:00"
    }
  ]

```

}

3.8.2 機能概要

日常所見情報 API サービスのユースケース図を図 3-47 に示す。

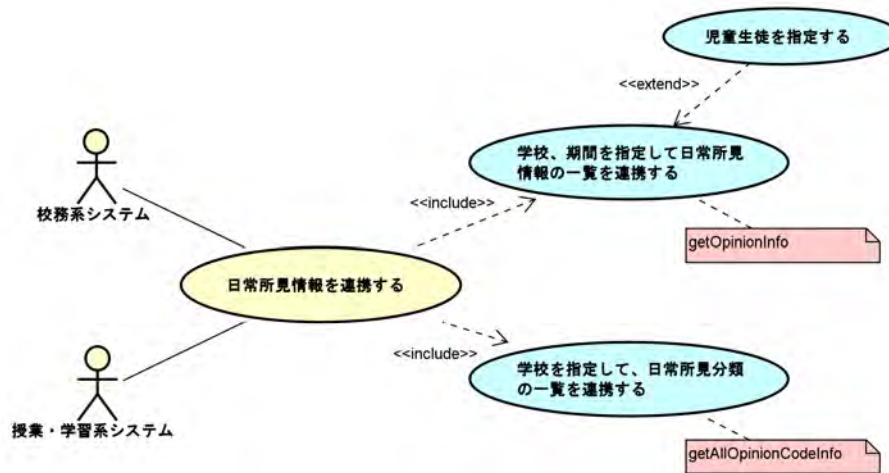


図 3-47 ユースケース図（日常所見情報 API サービス）

スマートスクール・プラットフォーム技術仕様では、日常所見情報について、各校務系システムもしくは授業・学習系システムで記録している児童生徒に関する日常生活や学習全般の様子等を日常所見として幅広い意味で用いる。また、日常所見以外の所見は仕様化対象外としている。

API サービス `getOpinionInfo` は、指定した暫定学校 ID (`temp_school_id`)、期間 (`since`、`until`)、システム連携用児童生徒共通 ID (`localstudent_id`) の日常所見情報の一覧を返す。指定した期間 (`since`、`until`) 内に 1 日でも在籍していた児童生徒が対象となる。

API サービス `getAllOpinionCodeInfo` で、指定した暫定学校 ID (`temp_school_id`) の児童生徒の所見分類コード情報の一覧を返す。

日常所見分類コードは、各校務系システムまたは授業・学習系システムで異なるため、各システムで定義している所見分類コードを API サービス `getAllOpinionCodeInfo` で連携させるようにしている。所見分類コードには、教科・学科コード、成績コード、時間割コードを使用することができ、独自に所見分類コードを定義することもできる。(所見分類コードの詳細については「3.1.4.2 教科・科目コードの利用範囲」を参照)

3.8.3 静的構造

日常所見情報のクラス図を図 3-48 に示す。

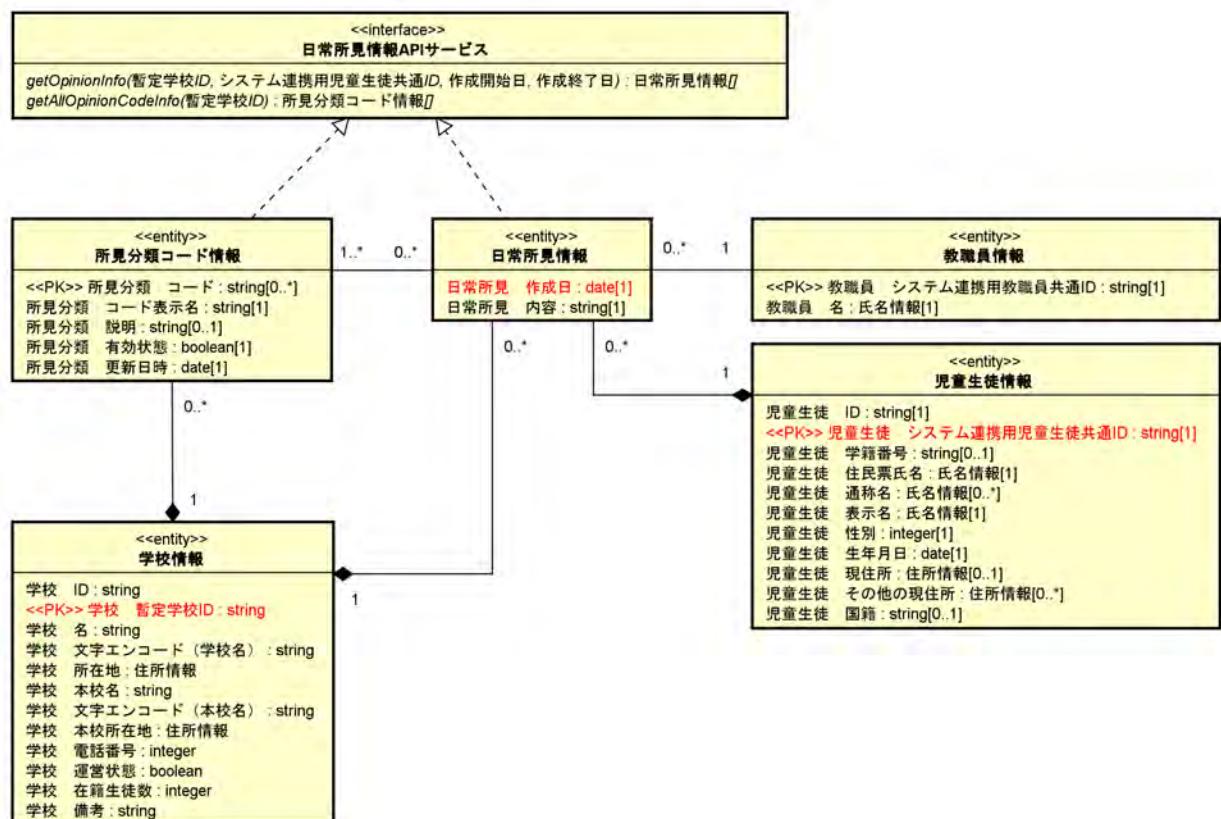


図 3-48 クラス図（日常所見情報 API サービス）

図 3-48 では、日常所見情報に児童生徒情報が関連付けられていることにより、API サービス `getOpinionInfo` で、対象児童生徒の日常所見情報を取得することができる。

同様に、所見分類コード情報には、学校情報が関連付けられていることにより、API サービス `getAllOpinionCodeInfo` で、対象学校の所見分類コード情報を取得することができる。

リクエストパラメータとクラス図との対応関係を表 3-49、表 3-50 に示す。

表 3-49 日常所見情報 API (`getOpinionInfo`) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID
localstudent_id	児童生徒情報	システム連携用児童生徒共通 ID
since	日常所見情報	作成日
until	日常所見情報	作成日

表 3-50 日常所見情報 API (getAllOpinionCodeInfo) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID

3.8.4 動的振る舞い

日常所見情報 API サービスのユースケース記述を表 3-51 に示す。ここでは例として、日常所見情報が校務系システムに登録されている前提で、それを授業・学習系システムに連携する場合について解説する。

表 3-51 日常所見情報 API サービスのユースケース記述

項目	内容
ユースケース名	日常所見情報を連携する。
概要	校務系システムに登録されている児童生徒の日常所見情報を、授業・学習系システムに連携する。
目的	(対応する文科省データ活用パターン) 生徒指導の充実 - データに基づいた生活指導 (担当教諭) - 児童生徒に対する組織的支援 (養護教諭、スクールカウンセラー、スクールソーシャルワーカー)
アクター	- 授業・学習系システム - 日常所見 API サービス (校務系システム) - 日常所見情報 (校務系システム) - 所見分類コード情報 (校務システム)
開始条件	(ユースケースを開始する条件。開始のきっかけ・トリガー) 授業・学習系システムから校務系システムの日常所見情報 API を呼び出す。
事前条件	(ユースケースが開始される前に満たされている条件) - 暫定学校 ID は校務系システムと授業・学習系システム間でマスタデータとして共有され、両システムに登録されていること。 - 授業・学習系システムと校務系システムに登録されている児童生徒情報の同期がとれていること。(「3.2 児童生徒情報 API サービス」参照)
正常フロー	(正常に処理が進んだ場合) - 授業・学習系システムは、学校、期間、児童生徒を指定して (児童生徒は任意)、校務系システムに日常所見情報を要求する。 - 校務系システムは、指定された学校、期間、児童生徒の日常所見情報を取得する。 - 校務系システムは、授業・学習系システムに、2 の日常所見情報を返す。 - 授業・学習系システムは、3 で受け取った日常所見情報を自システムに登録する。
代替フロー	(エラーが発生した場合。代替フローを実行しても事後条件は満たされる) 上記 2 で日常所見情報が 0 件の場合 (HTTP ステータスコード: 200)、ユースケ

項目	内容
	ースを終了する。
例外フロー	（ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない） 上記 2 で日常所見情報の取得に失敗した場合（HTTP ステータスコード：404）、ユースケースを中断する。
事後条件	（ユースケースが実行された後に満たされている条件） 授業・学習系システムと校務系システムに登録されている日常所見情報の同期がとれていること。

日常所見情報 API サービスのイベントフロー図を図 3-49 に示す。

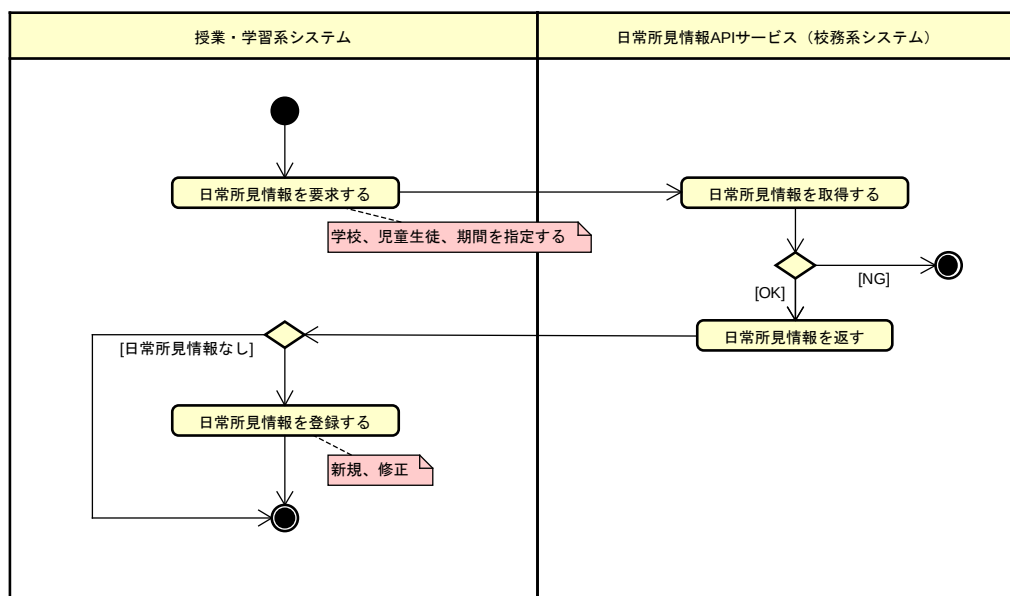


図 3-49 イベントフロー図（日常所見情報 API サービス）

暫定学校 ID は校務系システムと授業・学習系システムとの間で事前に共有されているものとする。また、システム連携用児童生徒共通 ID についても、児童生徒情報 API サービスによるデータ連携により、校務系システムと授業・学習系システムとの間で事前に共有されているものとする。

日常所見情報 API サービスのシーケンス図を図 3-50 に示す。

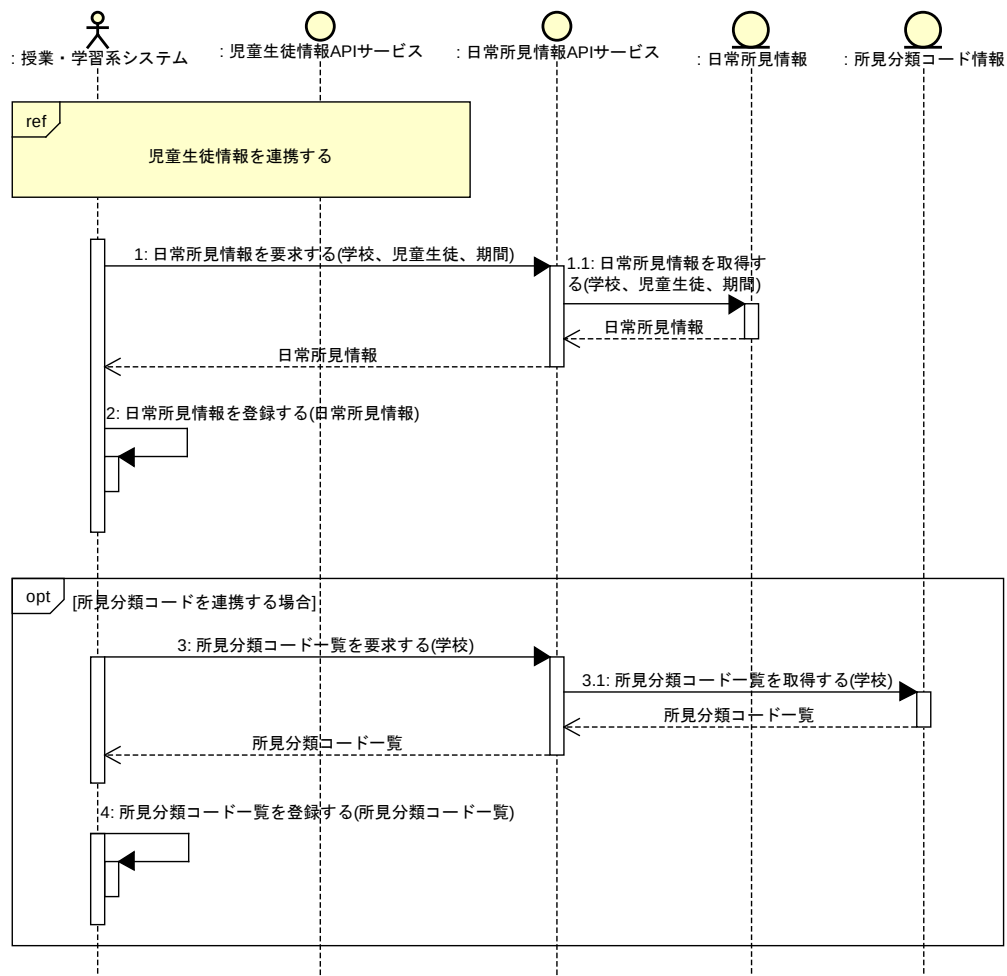


図 3-50 シーケンス図（日常所見情報 API サービス）

児童生徒の日常所見情報を取得するには、暫定学校 ID、期間、システム連携用児童生徒共通IDを指定してAPIサービス `getOpinionInfo` を呼び出すことにより（図中 1）、当該日常所見情報を取得し、授業・学習システムに登録する（図中 2）。

学校の所見分類コード情報の一覧を取得するには、暫定学校 ID を指定してAPIサービス `getAllOpinionCodeInfo` を呼び出すことにより（図中 3）、当該所見分類コード情報を取得し、授業・学習システムに登録する（図中 4）。

3.9 保健室利用情報 API サービス

3.9.1 仕様概要

3.9.1.1 getInfirmaryUseRecordInfo

(1) 説明

指定した学校、児童生徒、期間の保健室利用情報の一覧を返す。

(2) パス

GET /infirmary_use_record_info/{temp_school_id}/{localstudent_id}

(3) リクエストパラメータ

表 3-52 保健室利用情報 API (getInfirmaryUseRecordInfo) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
パス	temp_school_id	○	暫定学校 ID
パス	localstudent_id		システム連携用児童生徒共通 ID
クエリ	since	○	期間（開始日）(YYYY-MM-DD)
クエリ	until	○	期間（終了日）(YYYY-MM-DD)

(4) レスポンス

表 3-53 保健室利用情報 API (getInfirmaryUseRecordInfo) レスポンス

プロパティ名	多重度	説明
localstudent_id	1	システム連携用児童生徒共通 ID
date_of_use	1	利用日 (YYYY-MM-DD)
enter_time	0..1	入室時間 (YYYY-MM-DDThh:mm:ss+09:00)
leave_time	0..1	退出時間 (YYYY-MM-DDThh:mm:ss+09:00)
scene	0..1	学校場面コード
place	0..1	学校生活場所コード
medical_subject	0..1	診療科（内科：010、外科：110、その他：n/a）
symptom	0..*	症状コード
part	0..*	部位（HL7 v2 table 0550 (Body Parts)の規定に準ずる）
cause	0..*	保健室来室原因コード

プロパティ名	多重度	説明
treatment	0..*	救急処置実施内容コード
temperature	0..1	体温
pulse	0..1	脈拍
note	0..1	備考

(5) サンプル

① リクエスト

GET /infirmary_use_record_info/J_0662711456/S0000059?since=2018-07-01&until=2018-07-30

② レスポンス

```
{
  "results": [
    {
      "localstudent_id": "S0000059",
      "date_of_use": "2018-07-12",
      "enter_time": "2018-07-12T10:00:00+09:00",
      "leave_time": "2018-07-12T11:00:00+09:00",
      "scene": "5th_period",
      "place": "exptroom",
      "medical_subject": "000",
      "symptom": [
        "VFM8"
      ],
      "part": [
        "NIPPL"
      ],
      "cause": [
```

```
        "undernourished"
    ],
    "treatment": [
        "rest_on_bed"
    ],
    "temperature": "36.7",
    "pulse": "82",
    "note": ""
}
]
```

```
}
```

3.9.2 機能概要

保険室利用記録情報 API サービスのユースケース図を図 3-51 に示す。

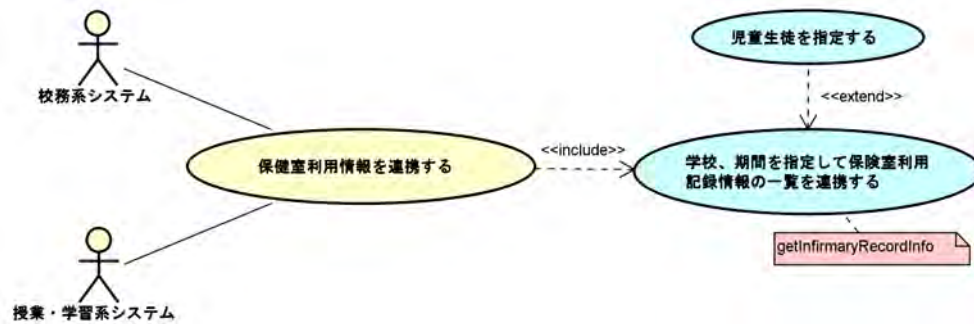


図 3-51 ユースケース図（保健室利用情報 API サービス）

API サービス `getInfirmaryUseRecordInfo` は、指定した暫定学校 ID (`temp_school_id`)、期間 (`since`、`until`) の保健室利用情報の一覧を返す。また、暫定学校 ID、期間と合わせて、システム連携用児童生徒共通 ID (`localstudent_id`) を指定することにより、特定の児童生徒の保健室利用情報の一覧を取得することができる。

3.9.3 静的構造

保健室利用情報のクラス図を図 3-52 に示す。

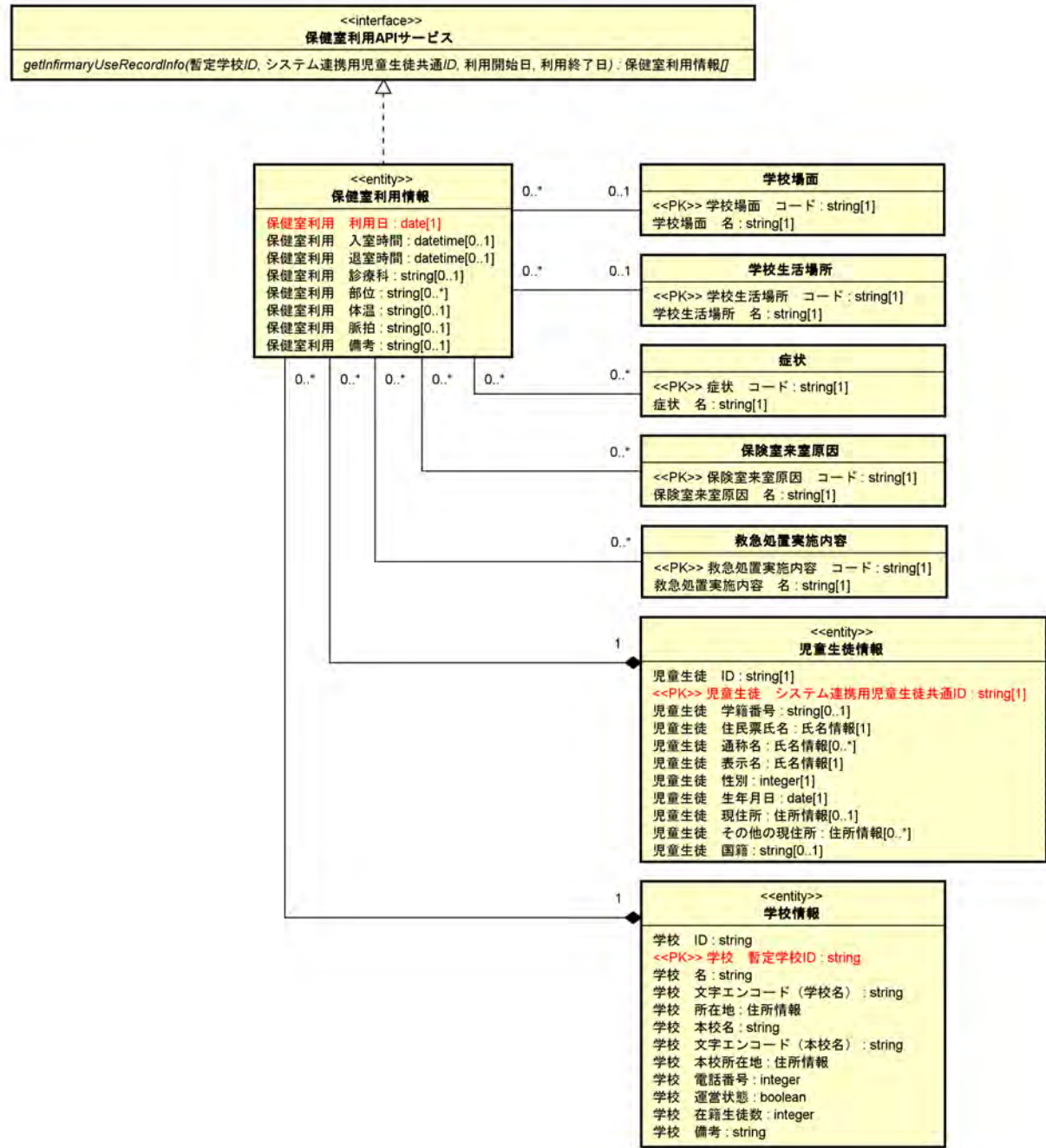


図 3-52 クラス図（保健室利用情報 API サービス）

図 3-52 では、保健室利用情報に児童生徒情報および学校情報が関連付けられていることにより、API サービス getInfirmaryUseRecordInfo で、対象の保健室利用情報を取得することができる。

リクエストパラメータとクラス図との対応関係を表 3-54 に示す。

表 3-54 保健室利用情報 API (getInfirmaryUseRecordInfo) クラス図対応表

パラメータ名	クラス名	プロパティ名
temp_school_id	学校情報	暫定学校 ID
localstudent_id	児童生徒情報	システム連携用児童生徒共通 ID
since	保健室利用情報	利用日
until	保健室利用情報	利用日

3.9.4 動的振る舞い

保健室利用情報 API サービスのユースケース記述を表 3-55 に示す。ここでは例として、保健室利用情報が校務系システムに登録されている前提で、それを授業・学習系システムに連携する場合について解説する。

表 3-55 保健室利用情報 API サービスのユースケース記述

項目	内容
ユースケース名	保健室利用情報を連携する。
概要	校務系システムに登録されている児童生徒の保健室利用情報を、授業・学習系システムに連携する。
目的	(対応する文科省データ活用パターン) 生徒指導の充実 - データに基づいた生活指導 (担当教諭) - 児童生徒に対する組織的支援 (養護教諭、スクールカウンセラー、スクールソーシャルワーカー)
アクター	- 授業・学習系システム - 保健室利用 API サービス (校務系システム) - 保健室利用情報 (校務系システム)
開始条件	(ユースケースを開始する条件。開始のきっかけ・トリガー) 授業・学習系システムから校務系システムの保健室利用情報 API を呼び出す。
事前条件	(ユースケースが開始される前に満たされている条件) - 暫定学校 ID は校務系システムと授業・学習系システム間でマスタデータとして共有され、両システムに登録されていること。 - 授業・学習系システムと校務系システムに登録されている児童生徒情報の同期がとれていること。(「3.2 児童生徒情報 API サービス」参照)
正常フロー	(正常に処理が進んだ場合) - 授業・学習系システムは、学校、期間、児童生徒を指定して (児童生徒は任意)、校務系システムに保健室利用情報を要求する。 - 校務系システムは、指定された学校、期間、児童生徒の保健室利用情報を取得する。 - 校務系システムは、授業・学習系システムに、2 の保健室利用情報を返す。 - 授業・学習系システムは、3 で受け取った保健室利用情報を自システムに登録する。
代替フロー	(エラーが発生した場合。代替フローを実行しても事後条件は満たされる) 上記 2 で保健室利用情報が 0 件の場合 (HTTP ステータスコード: 200)、ユースケースを終了する。
例外フロー	(ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない)

項目	内容
	上記 2 で保健室利用情報の取得に失敗した場合（HTTP ステータスコード：404）、ユースケースを中断する。
事後条件	（ユースケースが実行された後に満たされている条件） 授業・学習系システムと校務系システムに登録されている保健室利用情報の同期がとれていること。

保健室利用情報 API サービスのイベントフロー図を図 3-53 に示す。

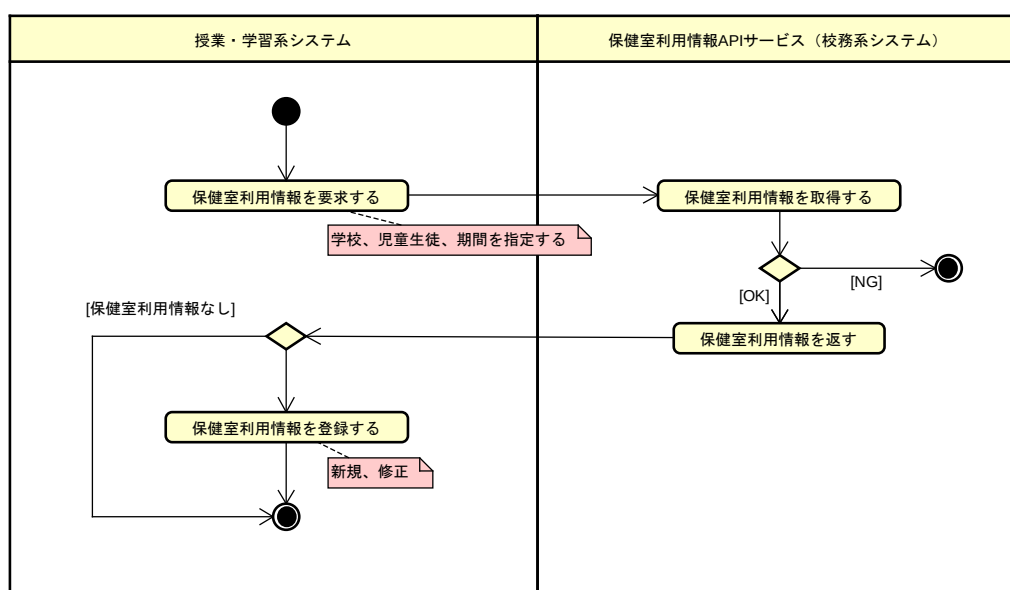


図 3-53 イベントフロー図（保健室利用情報 API サービス）

暫定学校 ID は校務系システムと授業・学習系システムとの間で事前に共有されているものとする。また、システム連携用児童生徒共通 ID についても、児童生徒情報 API サービスによるデータ連携により、校務系システムと授業・学習系システムとの間で事前に共有されているものとする。

保健室利用情報 API サービスのシーケンス図を図 3-54 に示す。

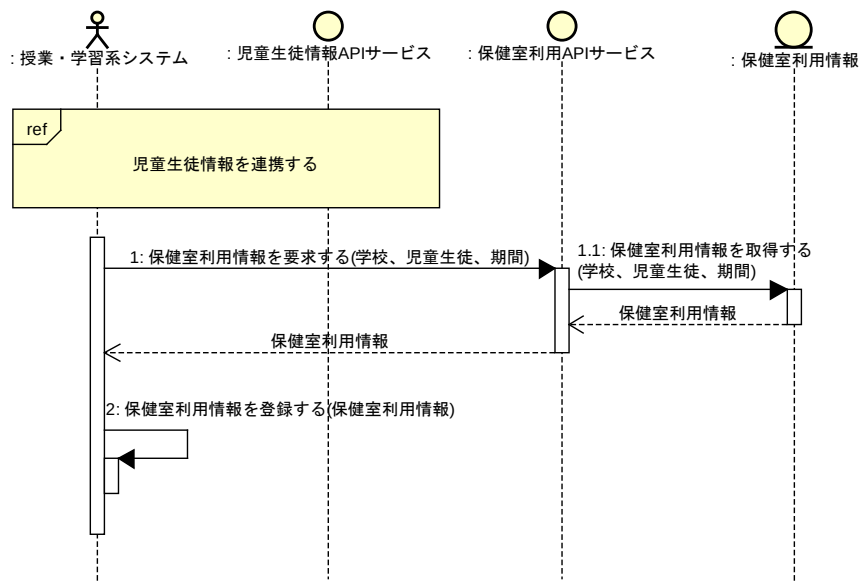


図 3-54 シーケンス図（保健室利用情報 API サービス）

保健室利用情報を取得するには、暫定学校 ID、期間、システム連携用児童生徒共通 ID を指定して API サービス `getInfirmaryUseRecordInfo` を呼び出すことにより（図中 1）、当該保健室利用情報を取得し、授業・学習系システムに登録する（図中 2）。

3.10 学習記録情報 API サービス

3.10.1 仕様概要

3.10.1.1 getStatementsInfoByLocalStudentId

(1) 説明

指定した児童生徒、教材、期間の学習記録情報ステートメントの一覧を返す。

(2) パス

GET /statements

(3) リクエストパラメータ

表 3-56 学習記録情報 API (getStatementInfoByLocalStudentId) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
クエリ	agent	○	児童生徒を表す agent オブジェクト
クエリ	activity		教材等を一意に表す activity_id
クエリ	since		期間（開始日時）
クエリ	until		期間（終了日時）

必要に応じて、xAPI の get ステートメントの仕様に記載された任意のリクエストパラメータを併用してもよい。

(4) レスポンス

「3.10.2.2 ステートメント構成」を参照。

3.10.1.2 getStatementsInfoWithAttachments

(1) 説明

指定した児童生徒、教材、期間の学習記録情報ステートメントの一覧と添付情報を返す。

(2) パス

GET /statements

(3) リクエストパラメータ

表 3-57 学習記録情報 API (getStatementInfoWithAttachments) リクエストパラメータ

ロケーション	パラメータ名	必須	説明
クエリ	agent	○	児童生徒を表す agent オブジェクト
クエリ	activity		教材等を一意に表す activity_id
クエリ	since		期間（開始日時）
クエリ	until		期間（終了日時）
クエリ	attachments		true 固定

必要に応じて、xAPI の get ステートメントの仕様に記載された任意のリクエストパラメータを併用してもよい。

(4) レスポンス

「3.10.2.2 ステートメント構成」を参照。

3.10.2 機能概要

3.10.2.1 基本フロー

スマートスクール・プラットフォーム技術仕様では、図 3-55 に示すように、学習記録情報の連携のユースケースとして、以下を想定している。

- ① テスト・ドリル、単問の学習記録情報を連携する
- ② アンケート回答、回答結果の分析情報を連携する

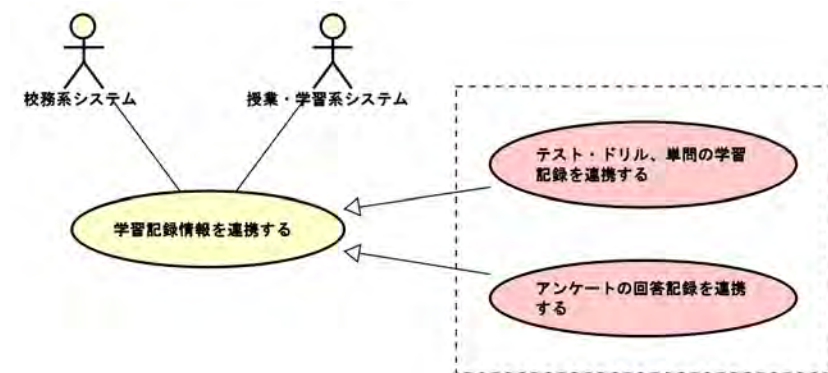


図 3-55 ユースケース図（学習記録情報 API サービス）（一部）

連携する学習記録情報は、一連の学習経験の記録情報（以下、「学習ログ情報」）から構成される。一連の学習経験の基本的なフローとして、上記のユースケース①、②の基本フローを、それぞれ「図 3-56 テスト・ドリル、単問の基本フロー」、「図 3-57 アンケートの基本フロー」に示す。

図 3-56、図 3-57 中の各アクション（「アプリケーションの起動」、「学習開始」など）ごとに学習ログ情報が存在する。

(1) テスト・ドリル、単問

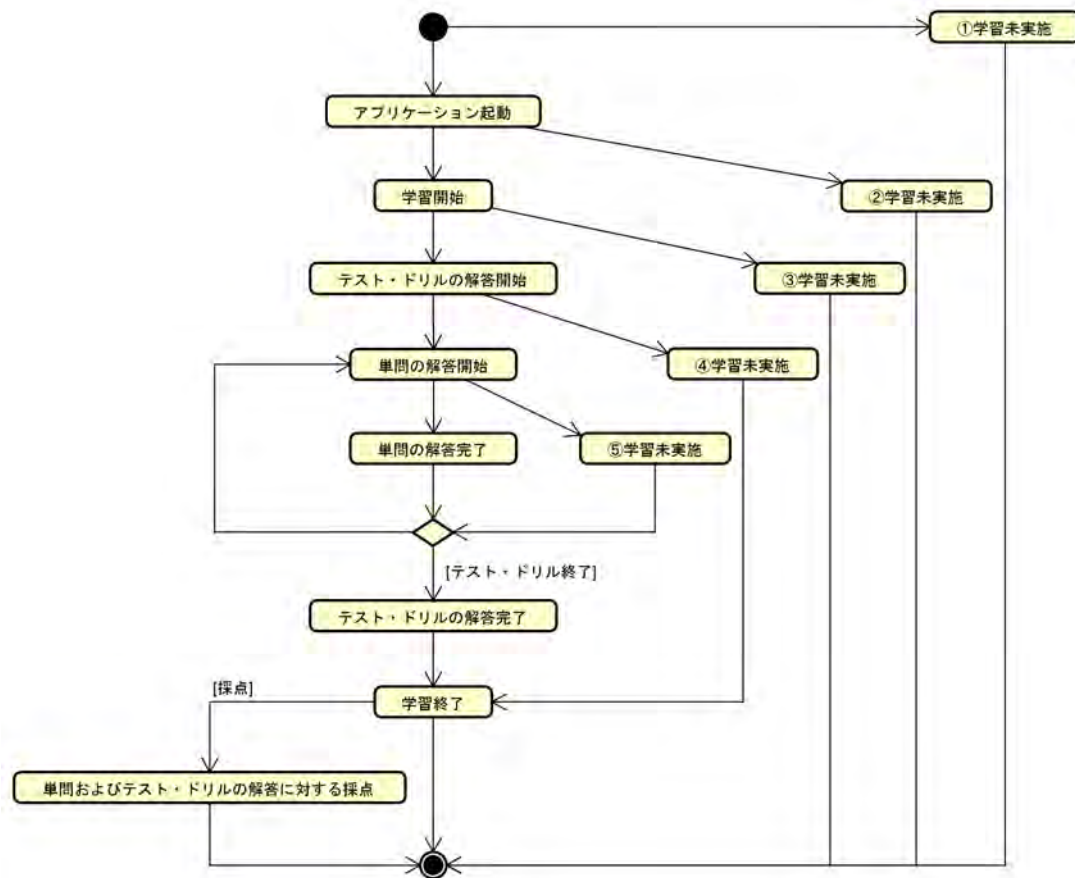


図 3-56 テスト・ドリル、単問の基本フロー

(2) アンケート

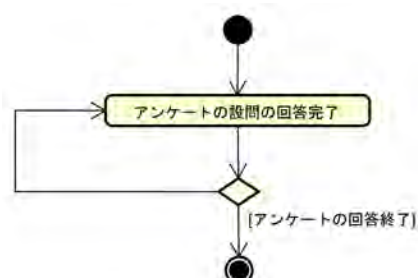


図 3-57 アンケートの基本フロー

3.10.2.2 ステートメント構成

学習ログ情報は、学習経験のシステム間通信に関する国際標準仕様である xAPI（「5.3 参考文献」参照）に準拠した形で連携させる。「図 3-56 テスト・ドリル、単問の基本フロー」、「図 3-57 アンケートの基本フロー」に示す児童生徒の一連の学習経験を、xAPI 仕様に準拠した一連のステートメント（以下、「ステートメント」）として表現する。

ステートメントの構成を「図 3-58 ステートメント全体構成（抜粋）」に示す。（「図 3-58」は便宜上、一部の構造、関連を省略している。ステートメントの全体構成の詳細については巻末付録「図 5-4 ステートメント全体構成」を参照）

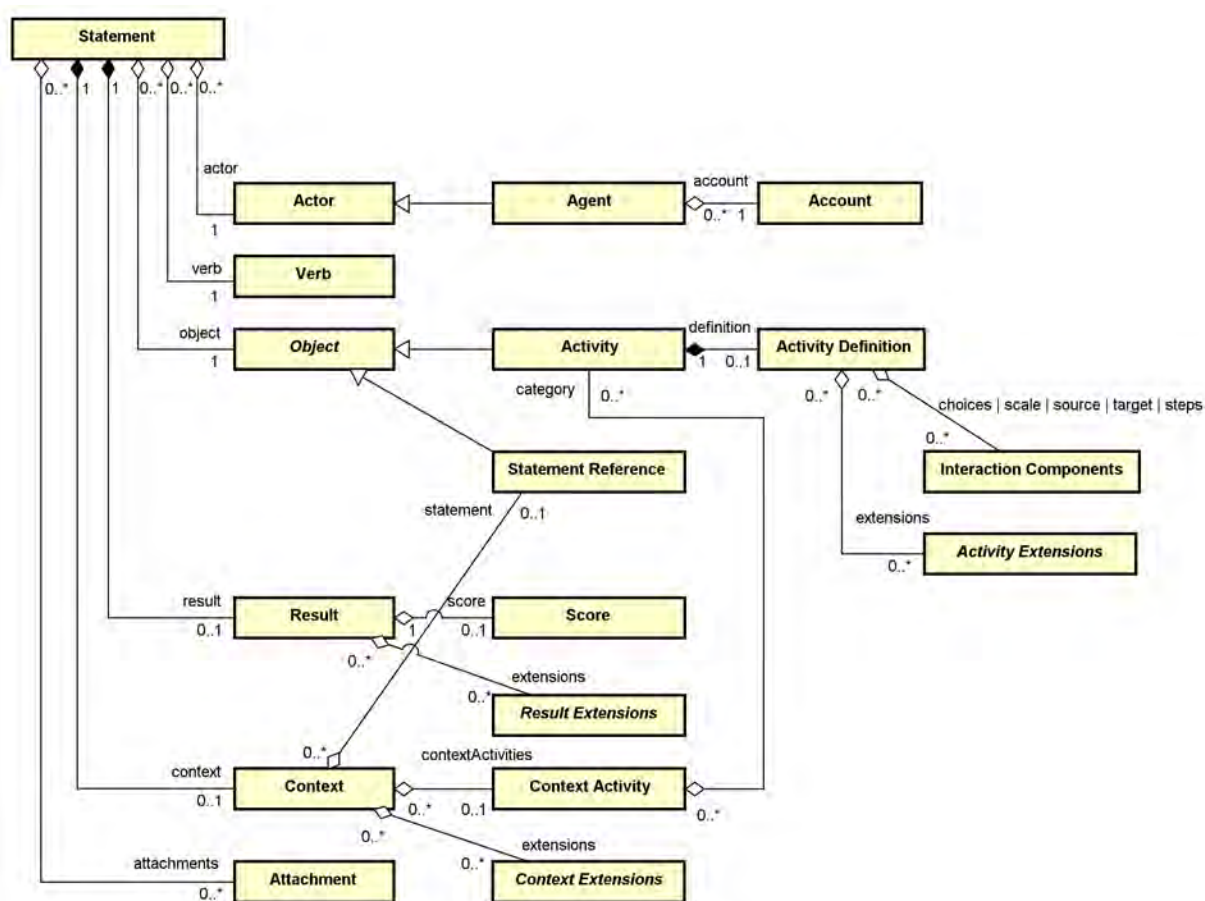


図 3-58 ステートメント全体構成（抜粋）

「図 3-56 テスト・ドリル、単問の基本フロー」、「図 3-57 アンケートの基本フロー」に示した基本フローのアクションひとつに、ステートメントが1:1で対応する。

各アクションのステートメント構成を以下に示す。

(1) テスト・ドリル、単問

① アプリケーション起動

Activity id には、学習コンテンツの製品・サービスを表す IRI を設定する。

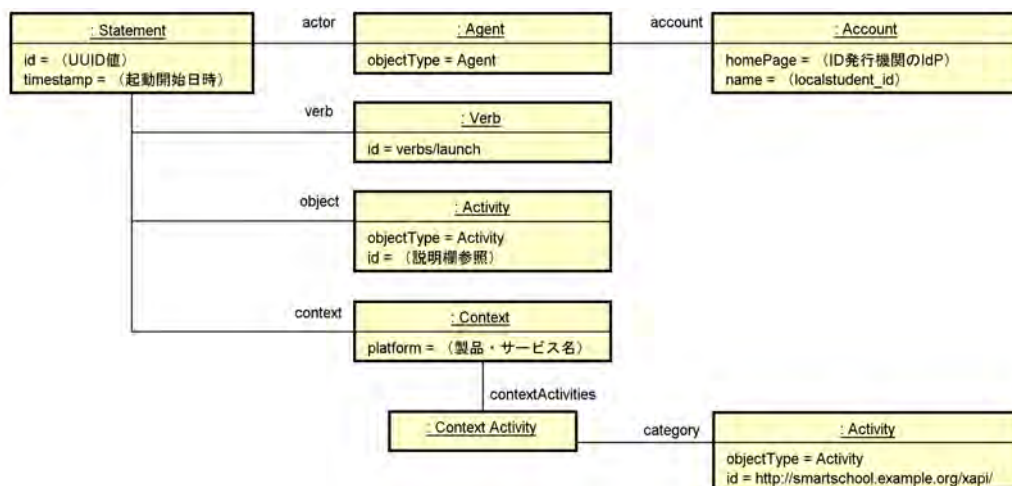


図 3-59 ステートメント構成（アプリケーション起動）

② 学習開始

Activity id には、コンテンツメタデータ（「5.1.6 結果データ」参照）のテストメタ ID を設定する。

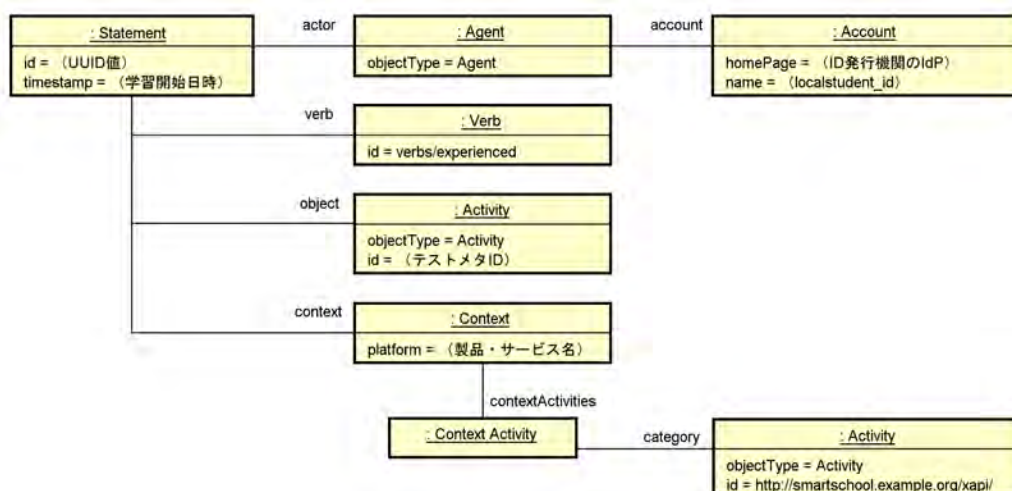


図 3-60 ステートメント構成（学習開始）

③ 学習終了

Activity id には、コンテンツメタデータ（「5.1.6 結果データ」参照）のテストメタ ID を設定する。

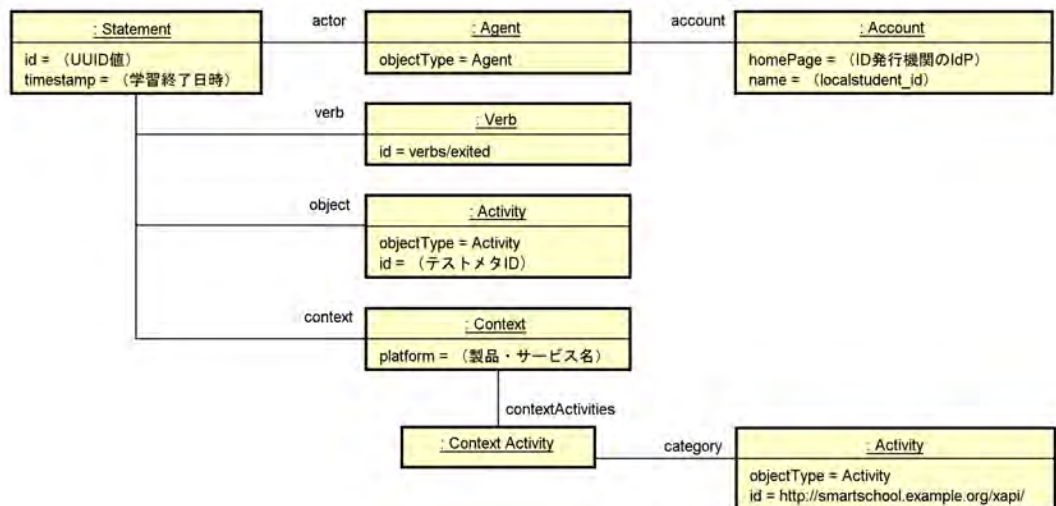


図 3-61 ステートメント構成（学習終了）

④ 学習未実施

Activity id は、「図 3-56 テスト・ドリル、単問の基本フロー」に示す使用場面により、以下のように設定する。

- ① 学習コンテンツの製品・サービスを表す IRI
- ② (同上)
- ③ テストメタ ID
- ④ (同上)
- ⑤ 単問メタ ID

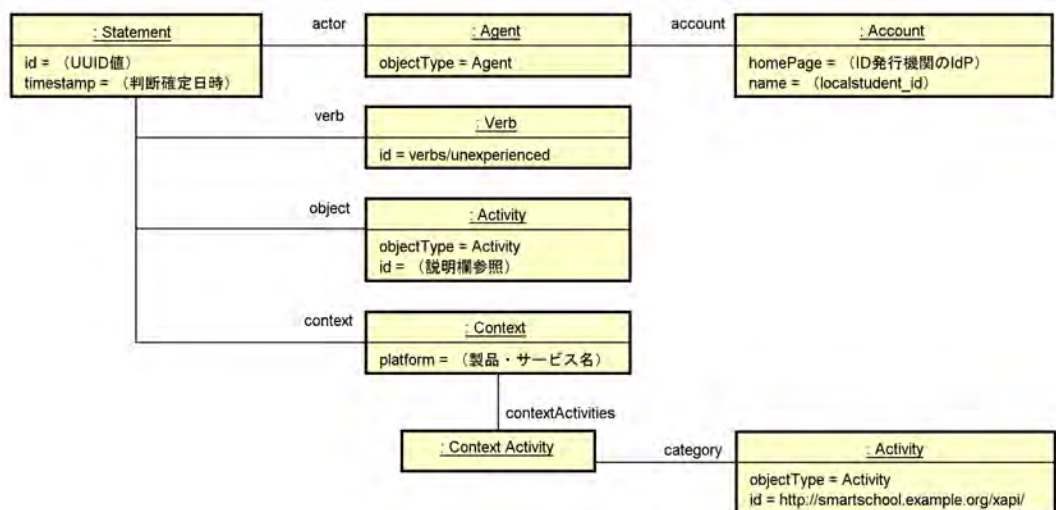


図 3-62 ステートメント構成（学習未実施）

⑤ 単問の解答開始

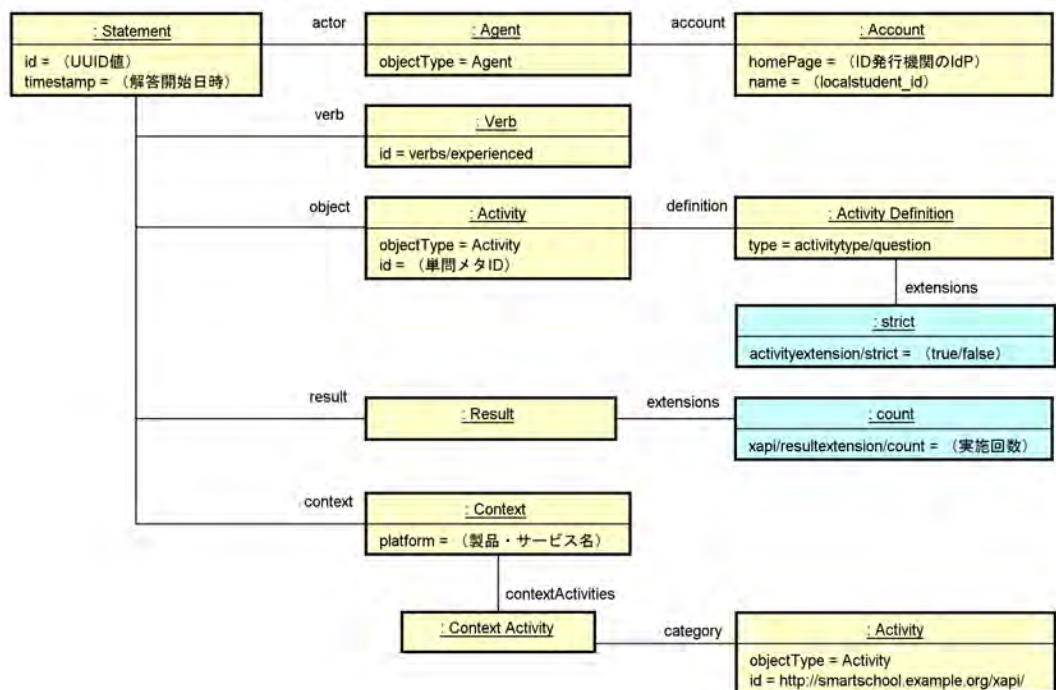


図 3-63 ステートメント構成（単問の解答開始）

⑥ 単問の解答完了

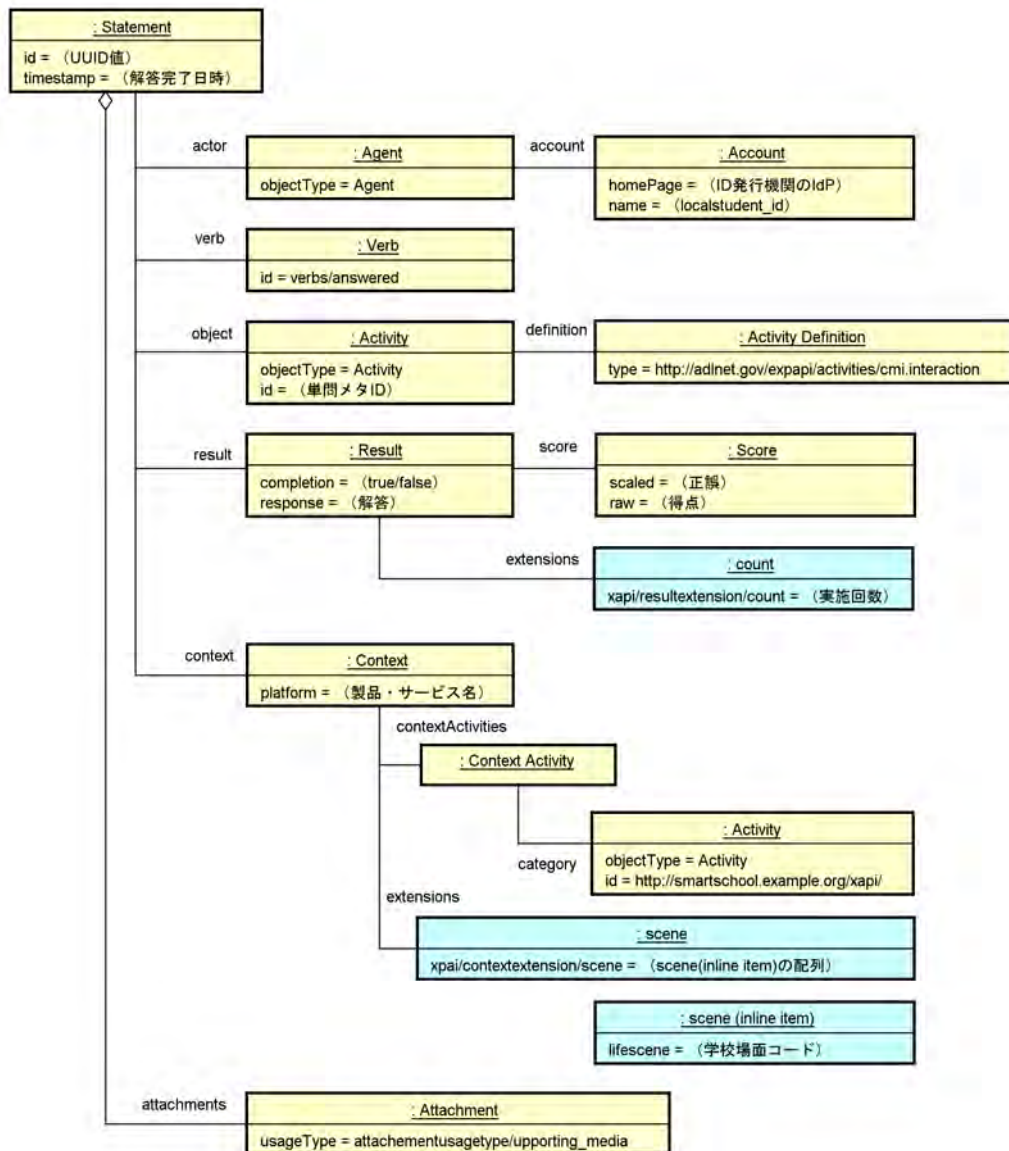


図 3-64 ステートメント構成（単問の解答終了）

⑦ テスト・ドリルの解答開始

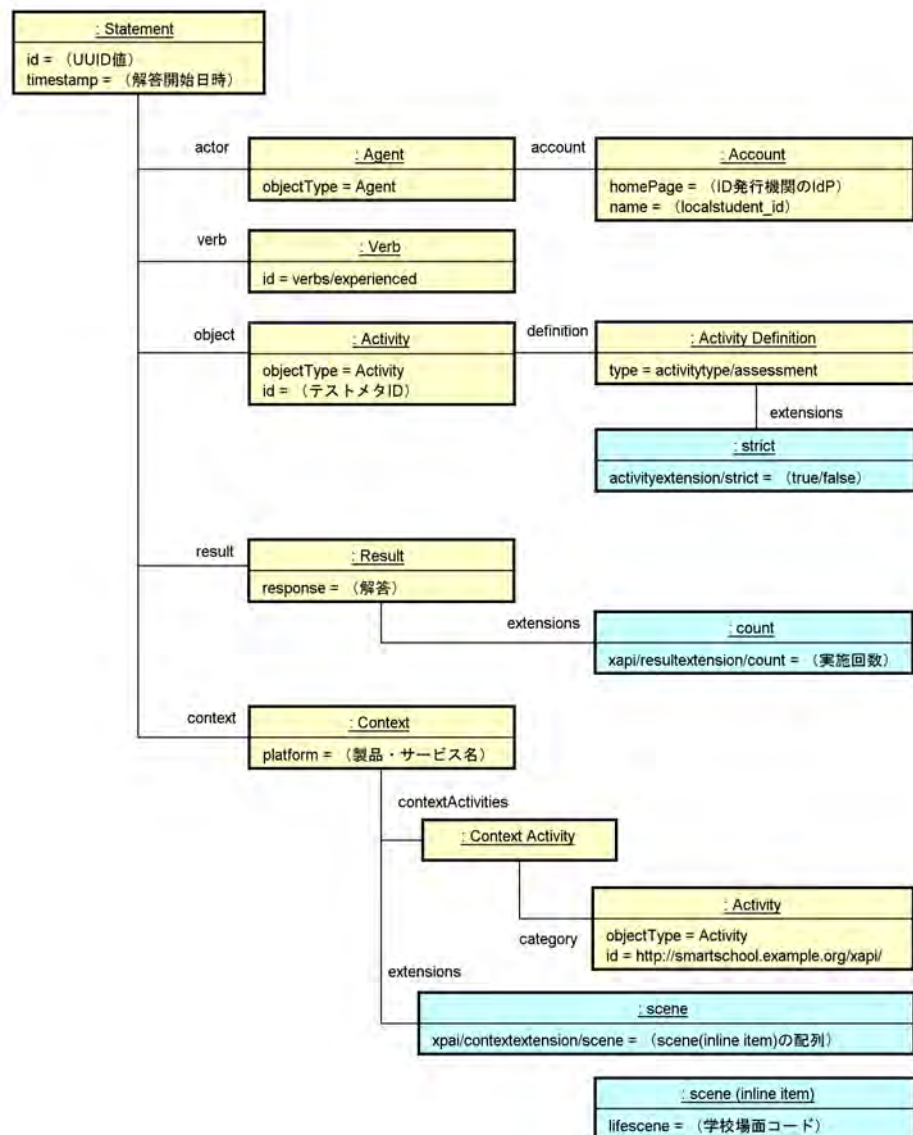


図 3-65 ステートメント構成 (テスト・ドリルの解答開始)

⑧ テスト・ドリルの解答完了

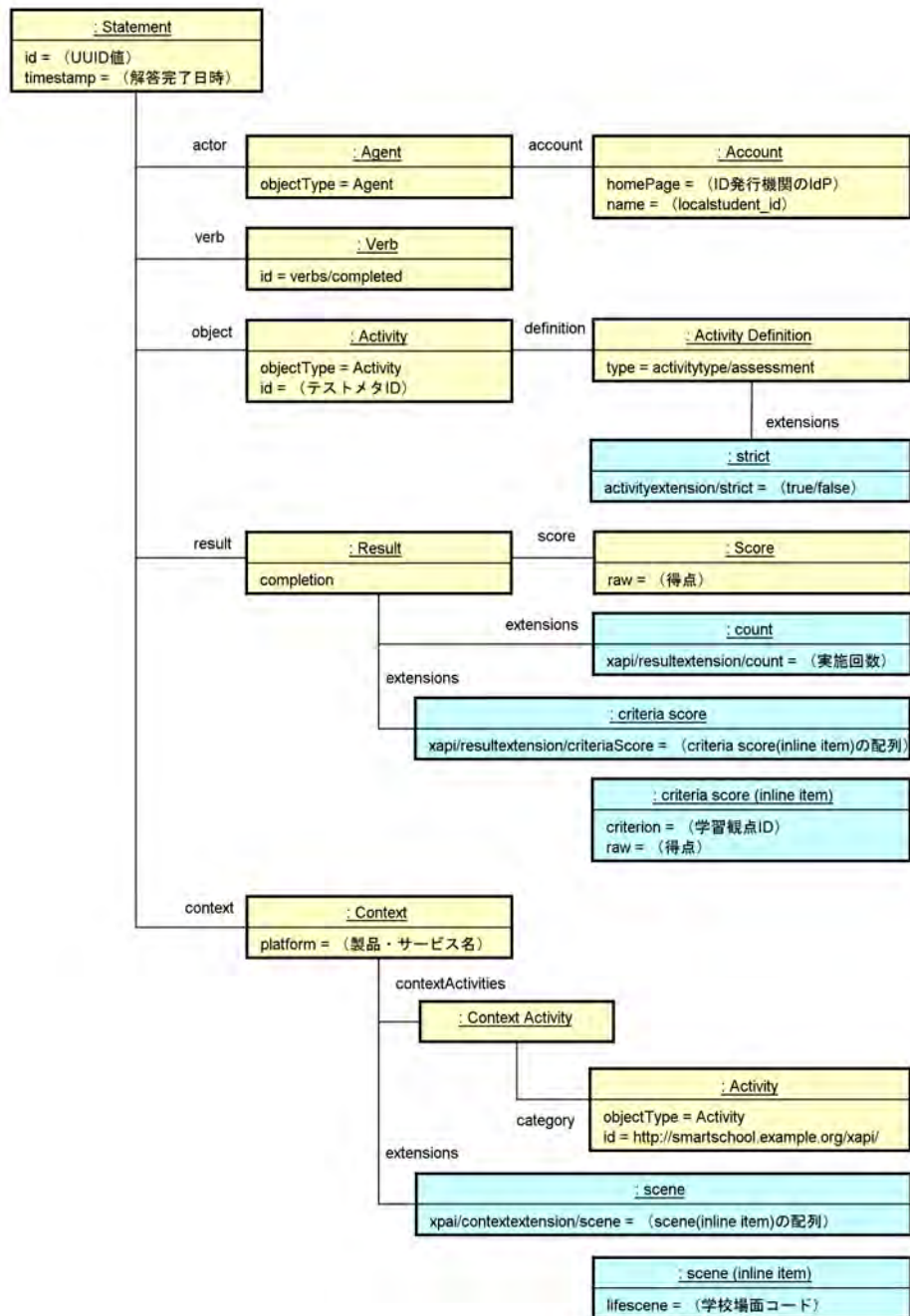


図 3-66 ステートメント構成 (テスト・ドリルの解答完了)

⑨ 単問およびテスト・ドリルの解答に対する採点

Statement Reference id には、採点対象となる「単問の解答完了」または「テスト・ドリルの解答完了」ステートメントの id を設定する。

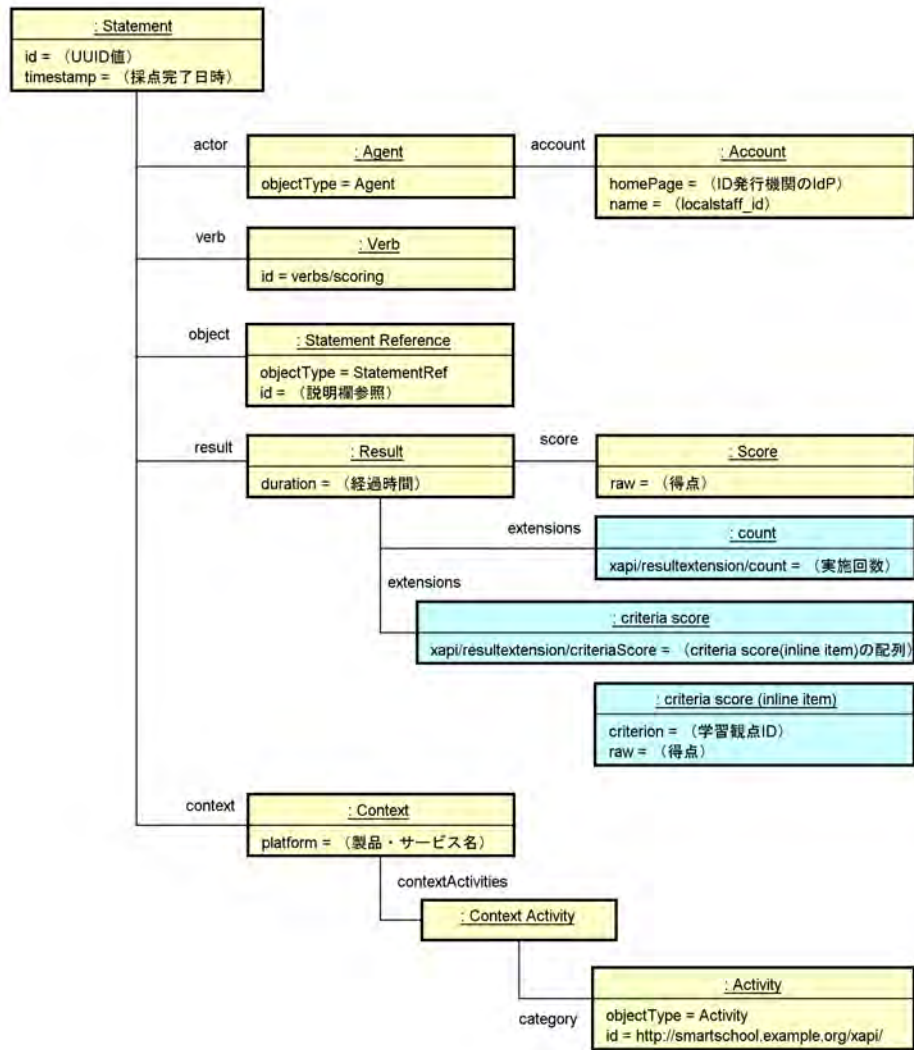


図 3-67 ステートメント構成（単問およびテスト・ドリルの解答に対する採点）

(2) アンケート

① 設問の回答完了

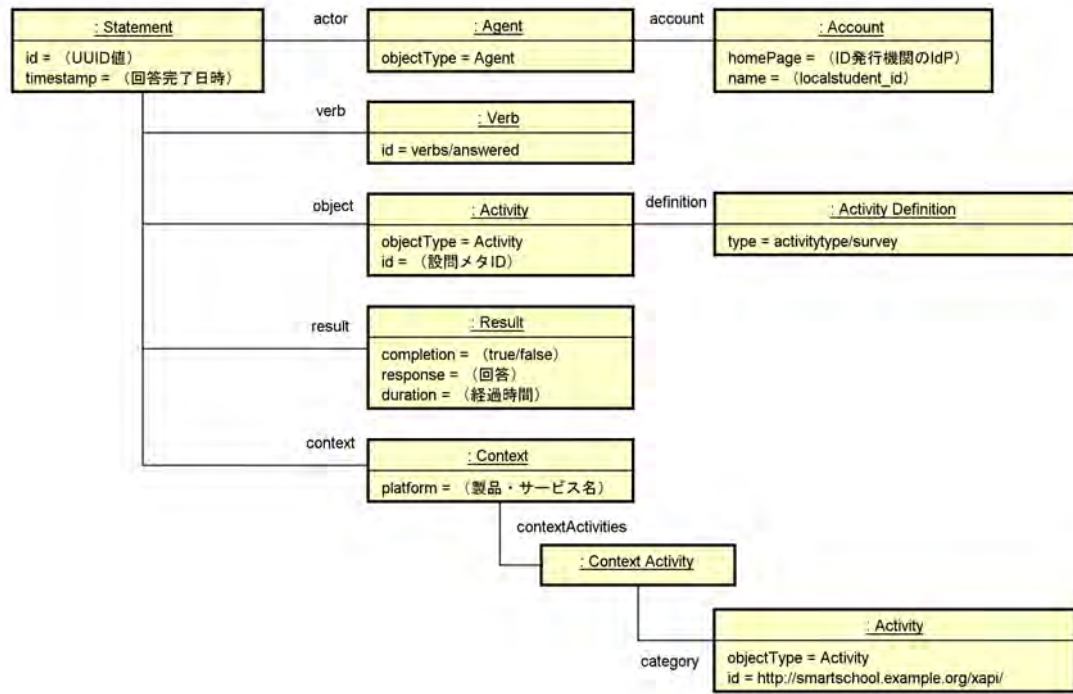


図 3-68 ステートメント構成（設問の回答完了）

3.10.2.3 ステートメント API

スマートスクール・プラットフォーム技術仕様では、学習記録情報を連携する API として、学習経験のシステム間通信に関する国際標準仕様である xAPI（「5.3 参考文献」参照）に準拠した以下の 2 つの API を規定している。

- ① `getStatementsInfoByLocalStudentId`
- ② `getStatementsInfoWithAttachments`

`getStatementsInfoByLocalStudentId` は、指定した児童生徒の学習記録情報の一覧を返す。

`getStatementsInfoWithAttachments` は、指定した児童生徒の学習記録情報の一覧を、解答内容を添付して返す。

上記 2 つの API とも、教材（activity）、期間（since、until）を指定することにより、学習記録情報を絞り込んで取得することができる。期間を指定するとその期間の間に実施されたテスト、ドリル等が取得できる。期間（since、until）には時刻情報も含まれる。

また、スマートスクール・プラットフォーム技術仕様では、必要に応じて xAPI の `get` ステートメントの仕様に記載された任意のリクエストパラメータを併用してもよい。

（例えば、単問解答結果のみを取得しようとした場合、リクエストパラメータとして、`verb` に“answered”を指定する。）

学習記録情報 API サービスの機能を図 3-69 に示す。

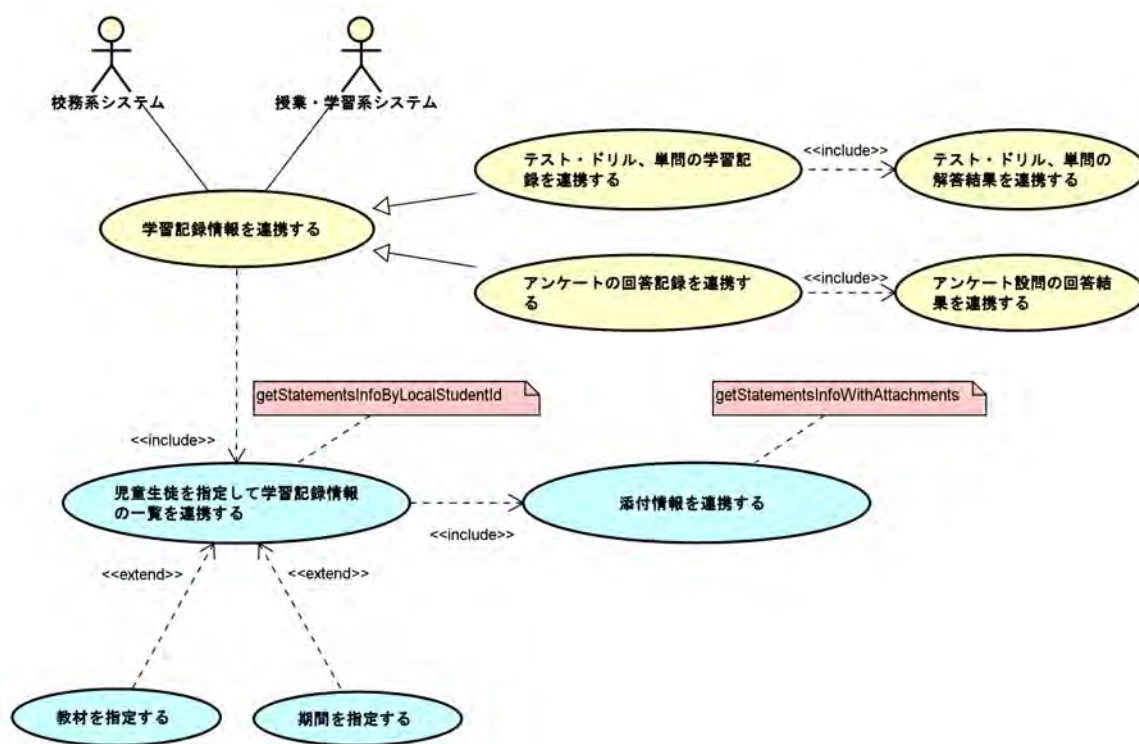


図 3-69 ユースケース図（学習記録情報 API サービス）

3.10.3 静的構造

学習記録情報 API サービスでは、前述した学習ログ情報の中に結果データを含めることにより、テスト・ドリルの解答結果、単問の解答結果、アンケート回答結果を連携することができる。（図 3-70）

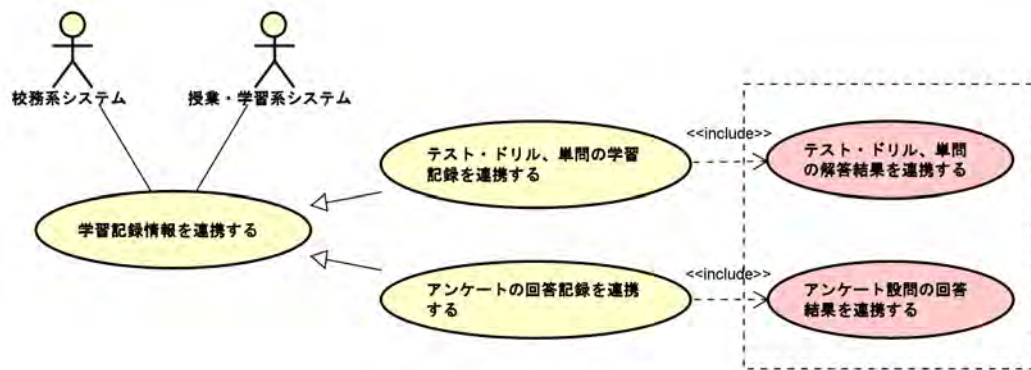


図 3-70 ユースケース図（学習記録情報 API サービス）（一部）

スマートスクール・プラットフォーム技術仕様では、テスト・ドリルの解答結果、単問の解答結果、アンケート回答結果（以下、「結果データ」）に対応するコンテンツメタデータを定義している。（図 3-71）

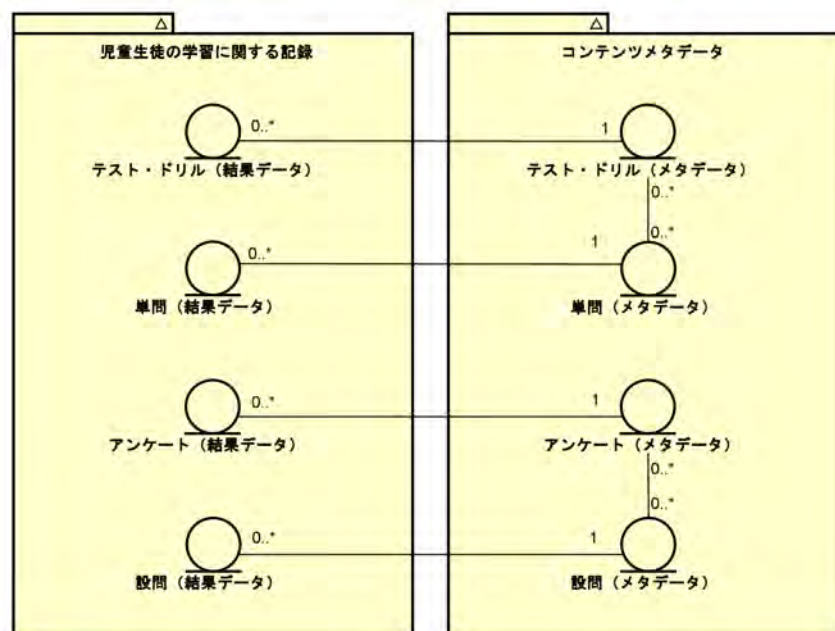


図 3-71 学習記録情報とコンテンツメタデータ（再掲）

3.10.3.1 結果データ

結果データとは、テスト・ドリルの解答結果、単問の解答結果、アンケート回答結果（「日時」、「実施回数」、「得点」など）を指す。結果データに対し、コンテンツメタデータは、テスト・ドリル、単問、アンケートのメタデータ（「テスト名」、「教科」、「単元」、「満点」など）を指す。コンテンツメタデータは、API サービスによる連携対象外であるため、連携するシステム間で事前に共有されていることを前提としている。

結果データは、テスト・ドリル、単問の基本フローの「テスト・ドリルの解答完了」、「単問の解答完了」に対応するステートメントの中に含まれている（図 3-72、図 3-73、図 3-74 参照）

前述したように、学習記録情報 API サービスのステートメント API は、必要に応じて xAPI の get ステートメントの仕様に記載された任意のリクエストパラメータを併用してもよい。リクエストパラメータの Verb にそれぞれ「verbs/completed」、「verbs/answered」を指定することにより、上記の「テスト・ドリルの解答完了」、「単問の解答完了」に対応するステートメントを取得することができる。

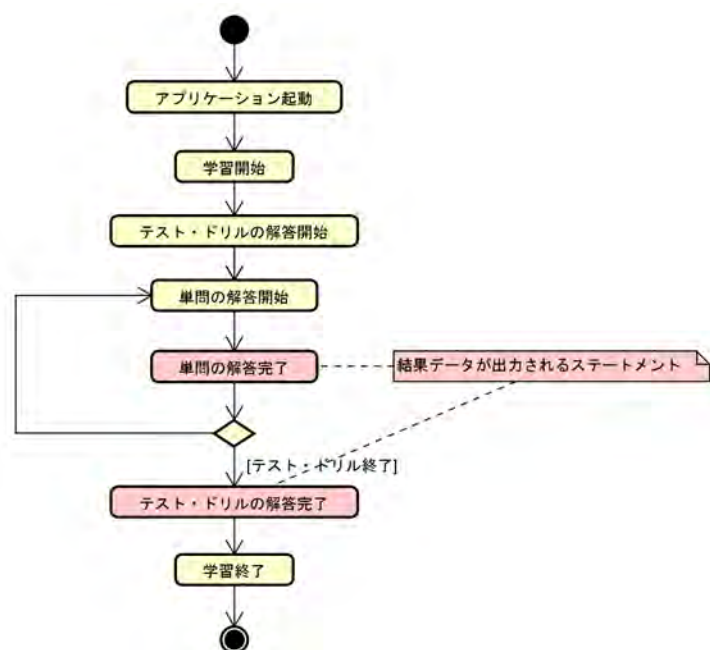


図 3-72 テスト・ドリル、単問の基本フロー（再掲）

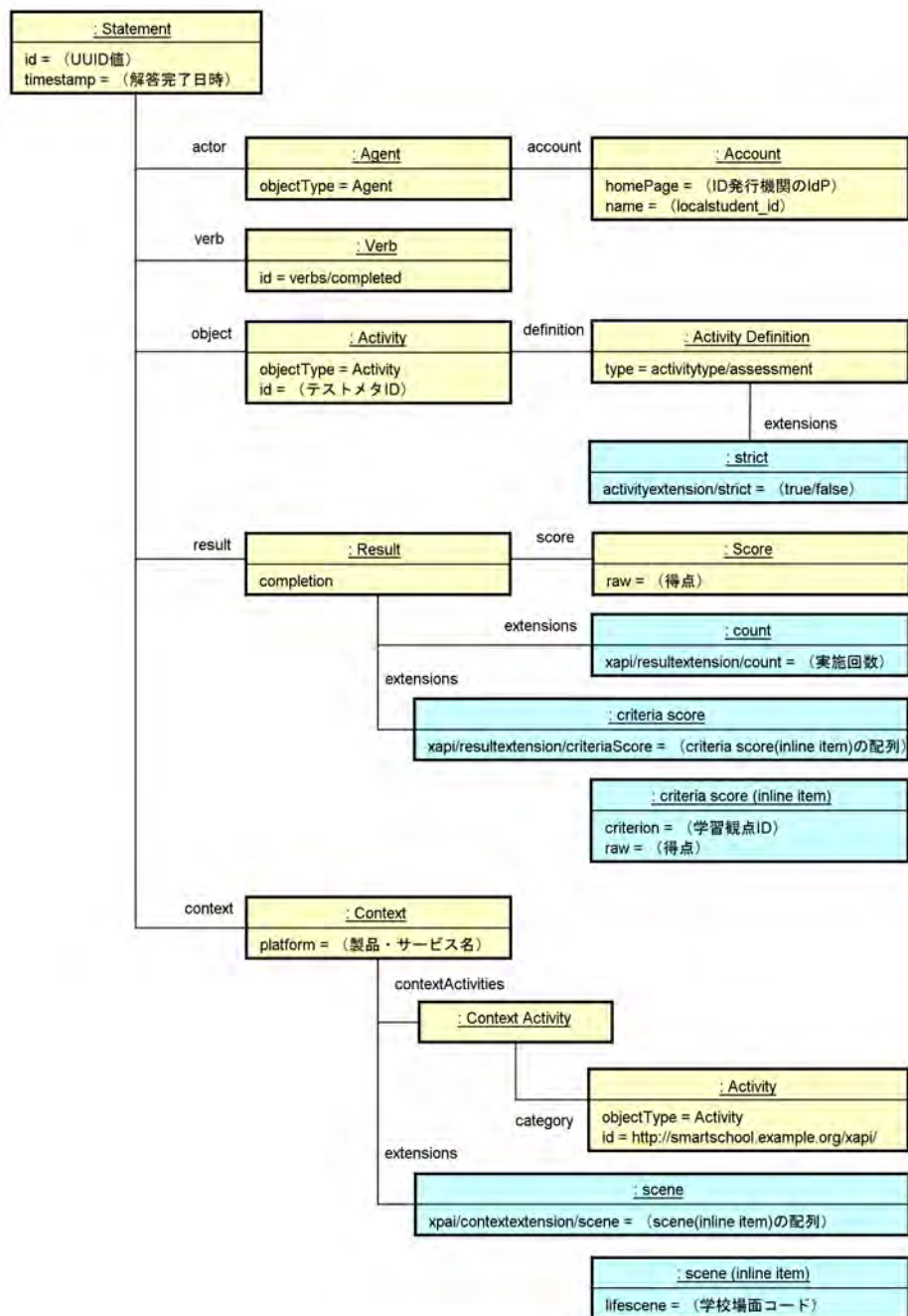


図 3-73 「テスト・ドリルの解答完了」ステートメント構成（再掲）

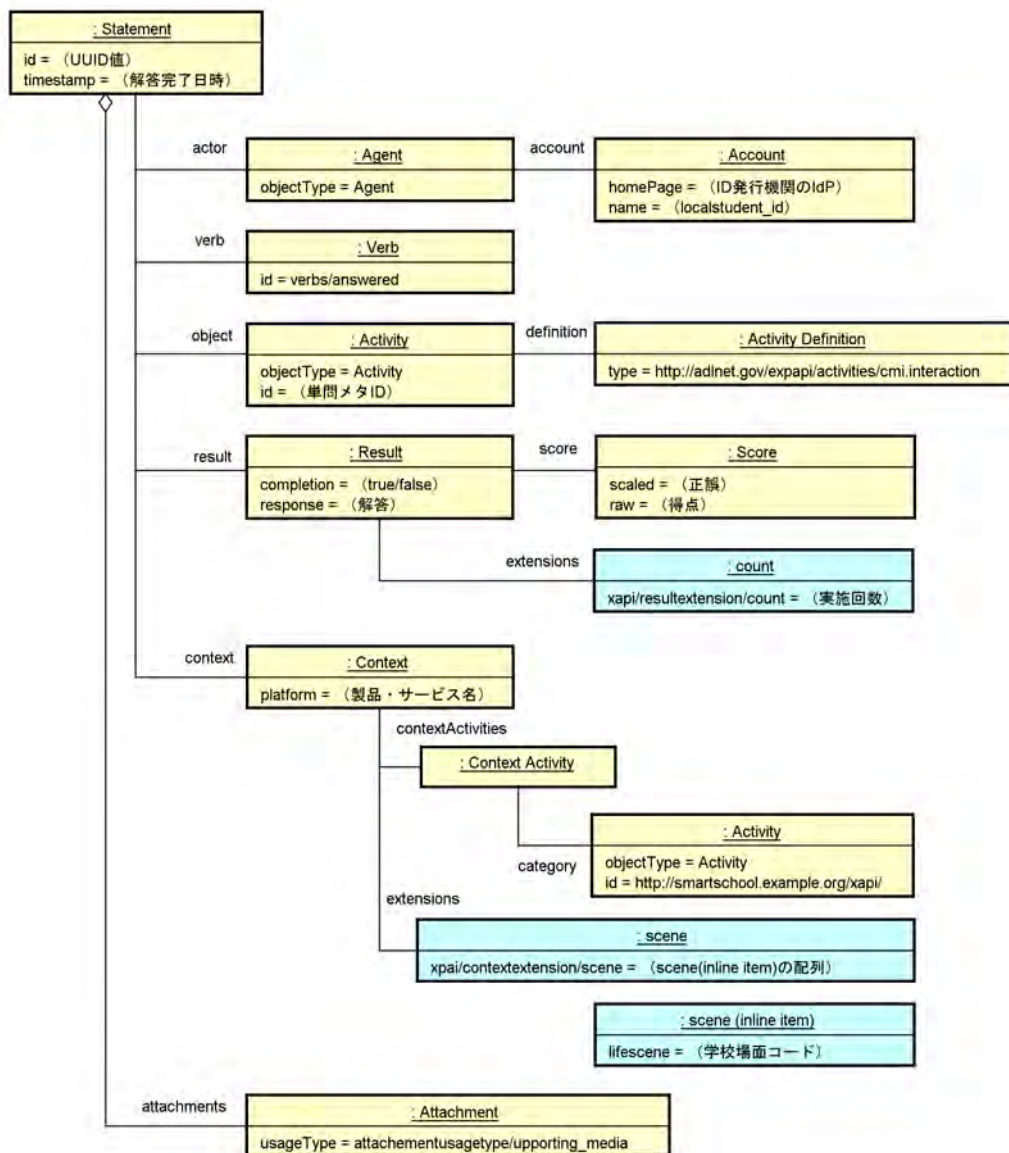


図 3-74 「単問の解答完了」テートメント構成（再掲）

3.10.3.2 結果データモデル

(1) テスト・ドリル、単問

学習記録情報（テスト・ドリル、単問）のクラス図を図 3-75 に示す。また、補足資料としてデータ項目定義を「5.1.5 コンテンツメタデータ」、「5.1.6 結果データ」に記載した。

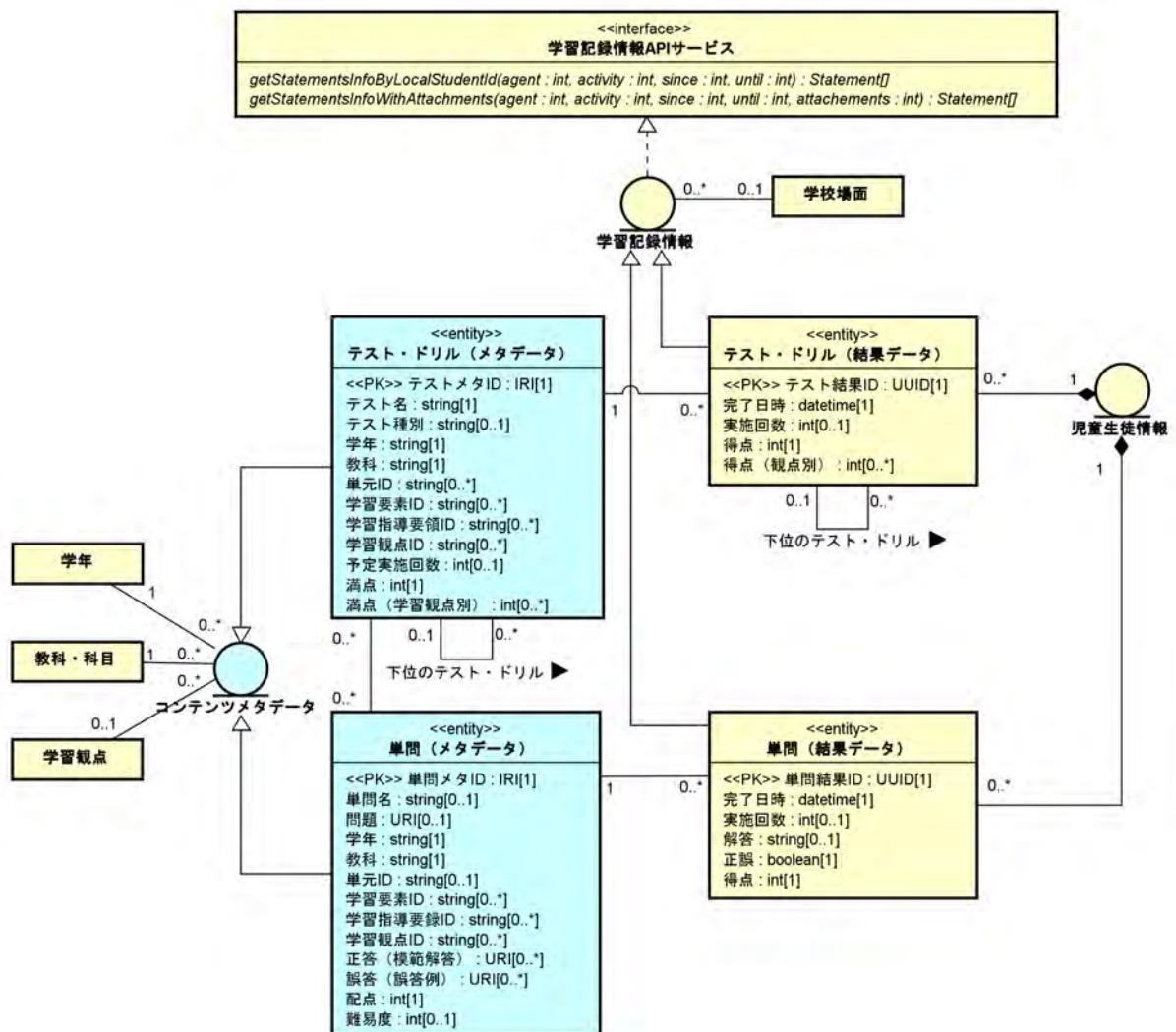


図 3-75 クラス図（学習記録情報 API サービス）（テスト・ドリル、単問）

(2) アンケート

学習記録情報（アンケート）のクラス図を図 3-76 に示す。また、補足資料としてデータ項目定義を「5.1.5 コンテンツメタデータ」、「5.1.6 結果データ」に記載した。

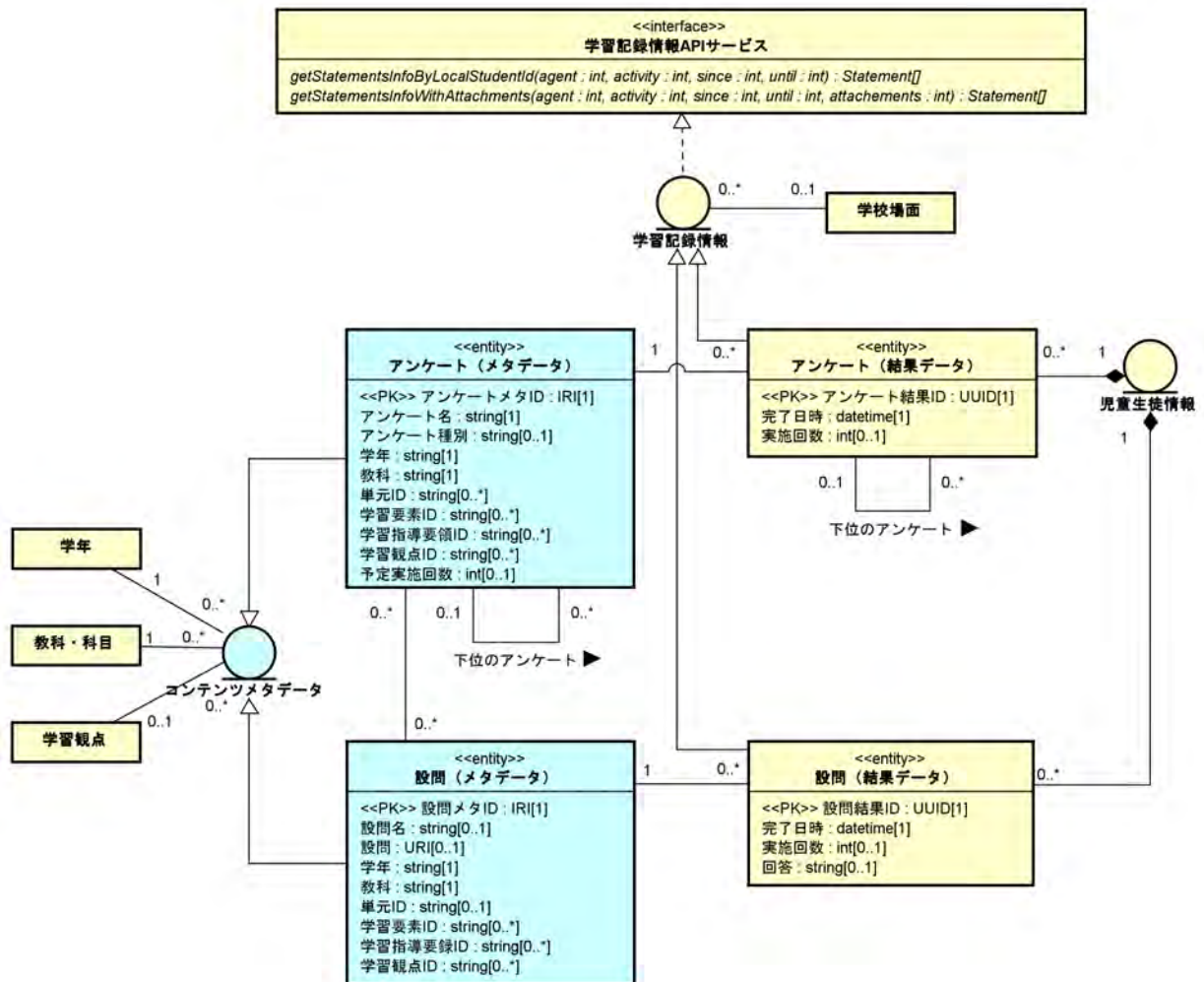


図 3-76 クラス図（学習記録情報 API サービス）（アンケート）

テスト・ドリル（メタデータ）は階層構造となっており、単問（メタデータ）は、階層構造の最下層という位置づけになっている。

テスト・ドリルの階層構造は、具体例として、定期テストの2学期期末テスト（親）に含まれる教科別テスト（子）（図 3-77 参照）、国語単元テスト（親）に含まれる各単元別テスト（子）（図 3-78 参照）、学習コンテンツの講座（親）に含まれる（講座）（図 3-79 参照）を表現する。

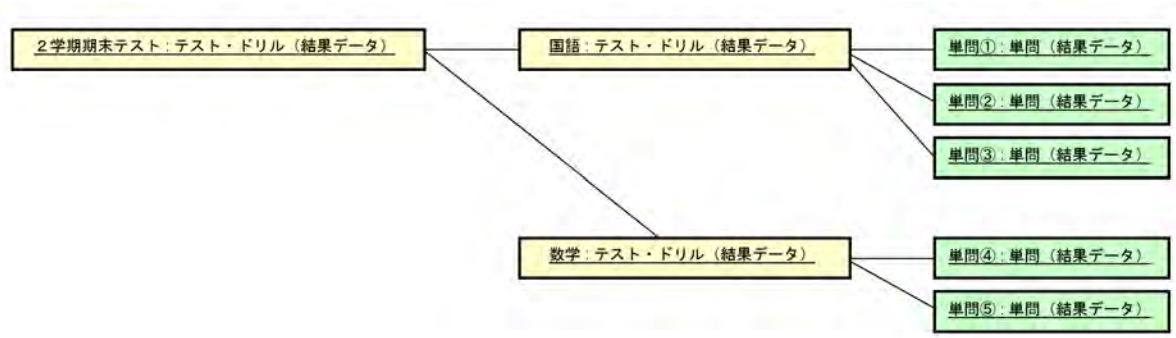


図 3-77 学習記録情報（テスト・ドリル、単問）データ例（定期テスト）

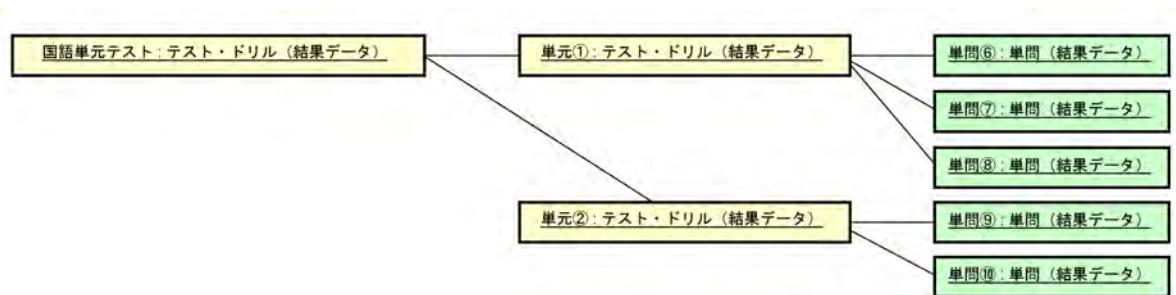


図 3-78 学習記録情報（テスト・ドリル、単問）データ例（単元テスト）

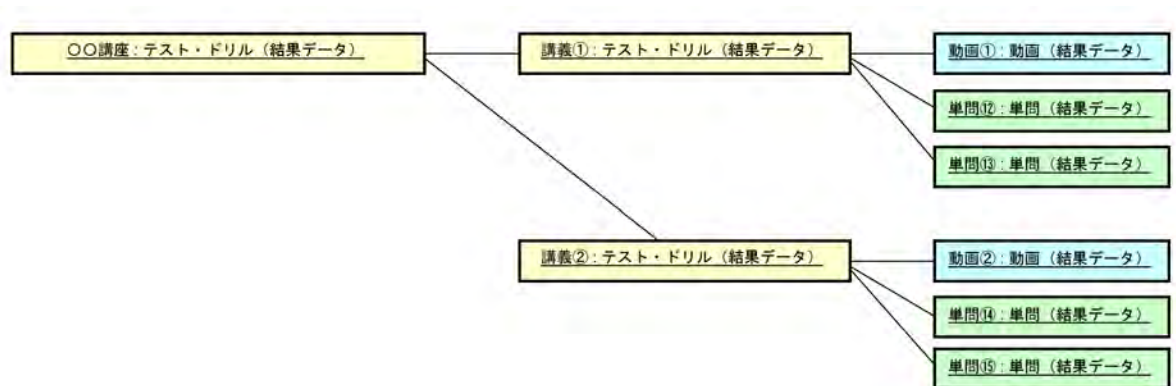


図 3-79 学習記録情報（テスト・ドリル、単問）データ例（講座）

3.10.4 動的振る舞い

学習情報記録 API サービスのユースケース記述を表 3-58 に示す。ここでは例として、学習記録情報が授業・学習系システムに登録されている前提で、それを校務系システムに登録する場合について解説する。

表 3-58 学習情報記録情報 API サービスのユースケース記述

項目	内容
ユースケース名	テスト・ドリル、単問の解答結果を連携する。
概要	授業・学習系システムに登録されている児童生徒のテスト・ドリル、単問の解答結果を、校務系システムに連携する。
目的	(対応する文科省データ活用パターン) 学習指導の充実 - 一人ひとりに応じた学習指導 - 授業改善
アクター	- 校務系システム - 学習記録情報 API サービス (授業・学習系システム) - 学習記録情報 (授業・学習系システム)
開始条件	(ユースケースを開始する条件。開始のきっかけ・トリガー) 授業・学習系システムから校務系システムの学習記録情報 API を呼び出す。
事前条件	(ユースケースが開始される前に満たされている条件) - 暫定学校 ID は校務系システムと授業・学習系システム間でマスタデータとして共有され、両システムに登録されていること。 - コンテンツメタデータは校務系システムと授業・学習系システム間でマスタデータとして共有され、両システムに登録されていること。 - 授業・学習系システムと校務系システムに登録されている児童生徒情報の同期がとれていること。(「3.2 児童生徒情報 API サービス」参照)
正常フロー	(正常に処理が進んだ場合) - 校務系システム B は、児童生徒、教材 (テスト・ドリル、単問)、期間を指定して (教材、期間は任意)、授業・学習系システム A に、テスト・ドリル、単問の解答結果を要求する。 - 授業・学習系システム A は、指定された児童生徒、教材 (テスト・ドリル、単問)、期間の解答結果を取得する。 - 授業・学習系システム A は、校務系システム B に 2 のテスト・ドリル、単問の解答結果を返す。 - 校務系システム B は、3 で受け取ったテスト・ドリル、単問の解答結果を自システムに登録する。
代替フロー	(エラーが発生した場合。代替フローを実行しても事後条件は満たされる)

項目	内容
	上記 2 で、テスト・ドリル、単問の解答結果が 0 件の場合 (HTTP ステータスコード: 200)、ユースケースを終了する。
例外フロー	(ユースケースの実行を中止する場合。例外フローを実行した場合、事後条件は満たされない) (なし)
事後条件	(ユースケースが実行された後に満たされている条件) 授業・学習系システムと校務系システムに登録されている児童生徒のテスト・ドリル、単問の解答結果の同期がとれていること。

学習記録情報 API サービスのイベントフロー図を図 3-80、図 3-81 に示す。

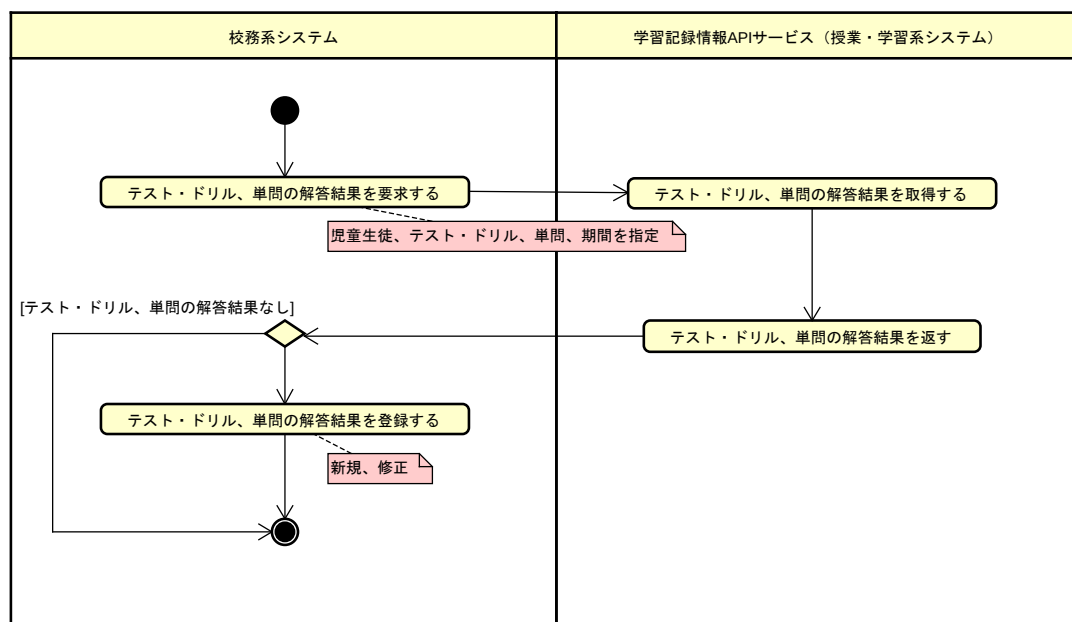


図 3-80 イベントフロー図 (学習記録情報 API サービス) (テスト・ドリル、単問)

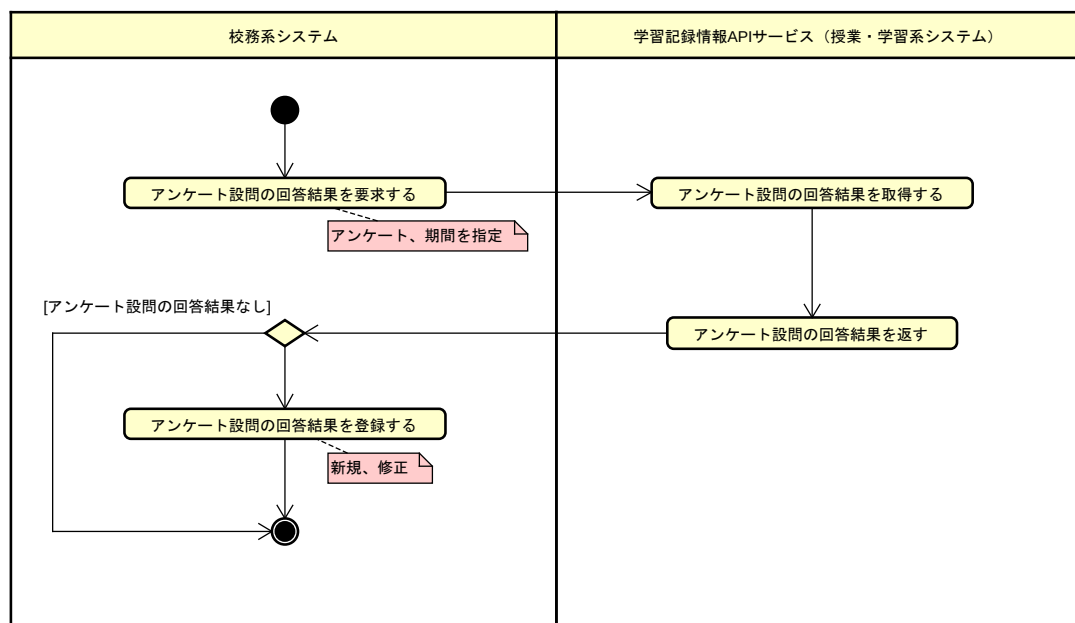


図 3-81 イベントフロー図（学習記録情報 API サービス）（アンケート設問）

暫定学校 ID およびコンテンツメタデータは、校務系システムと授業・学習系システムとの間で事前に共有されているものとする。コンテンツメタデータ管理(登録、更新、削除、問い合わせ応答等)は、スマートスクール・プラットフォーム技術仕様に規定する API を用いて学習記録情報を連携する場合、推奨構成とする。なお、スマートスクール・プラットフォーム技術仕様では、コンテンツメタデータのデータモデルのみ規定しており（「5.1.6 結果データ」参照）、その管理および連携方法等については特に規定していない。

また、システム連携用児童生徒共通 ID および学級については、それぞれ児童生徒情報 API サービス、学級情報 API サービスによるデータ連携により、校務系システムと授業・学習系システムとの間で事前に共有されているものとする。

学習記録情報 API サービスのシーケンス図を図 3-82 に示す。

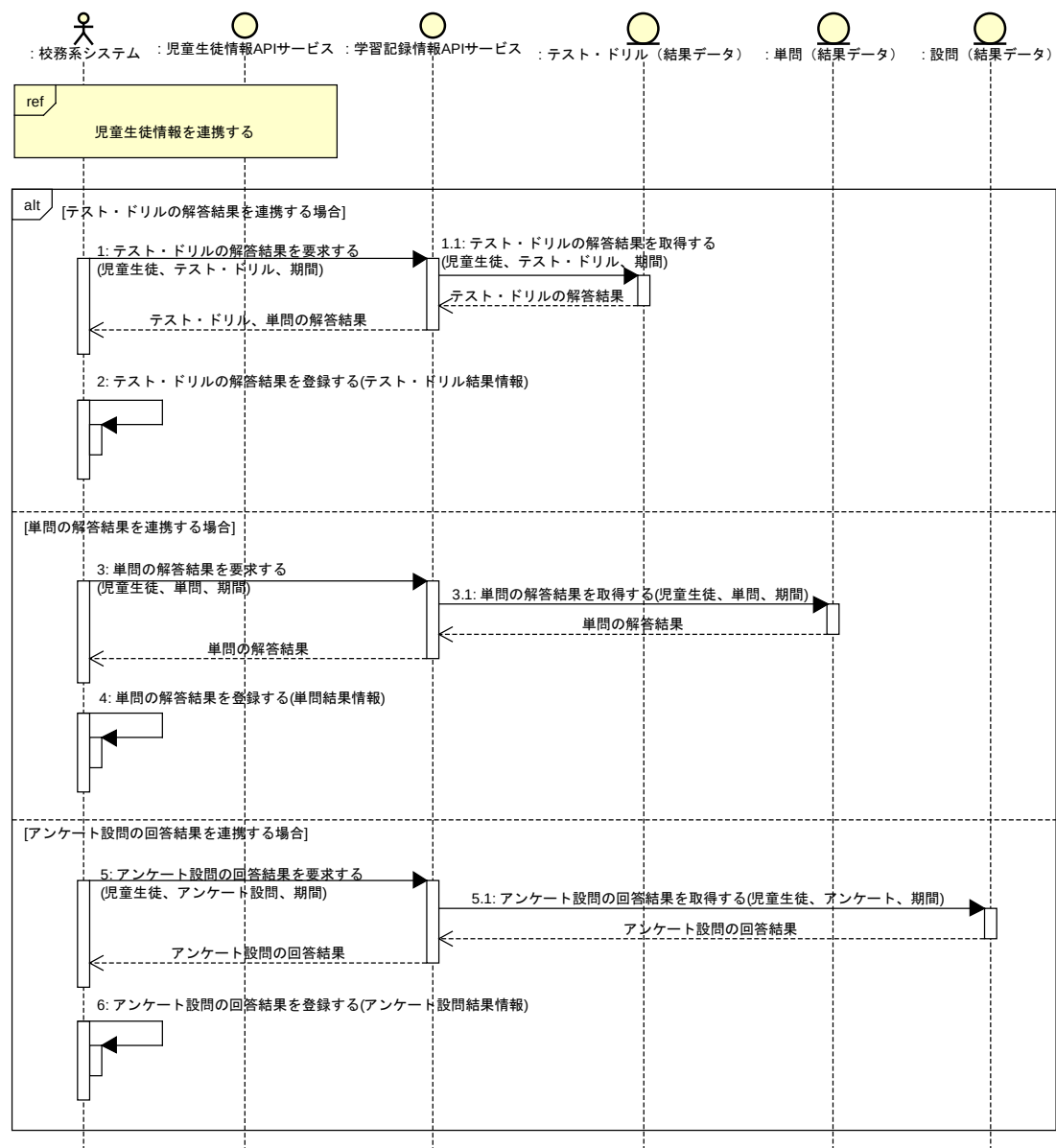


図 3-82 シーケンス図（学習記録情報 API サービス）

最初に API サービス `getStatementsInfoByLocalStudentId`（または、`getStatementsInfoWithAttachments`）で対象となる学習記録情報の一覧（Statement の配列）を取得する。（図中 1、3、5）

取得した学習記録情報を、校務系システムに登録する。（図中 2、4、6）

3.10.5 ケーススタディ

3.10.5.1 テスト・ドリル、単問の解答結果を連携する

ここではテスト・ドリルの例として、図 3-83 にあるような小学第3学年算数の単元テストを用いる。

スマスクドリル① きほん		学習日	年	月	日
単元	年 組 番	氏名			
3年「九九をおぼえよう」					
		22問			

1. 次の計算をしましょう。

(1) $2 \times 9 = \square$

(3) $8 \times 7 = \square$

(5) $3 \times 2 = \square$

(7) $0 \times 1 = \square$

(2) $5 \times 3 = \square$

(4) $4 \times 9 = \square$

(6) $2 \times 7 = \square$

(8) $4 \times 0 = \square$

2. $\square$ にあてはまる数や言葉を書きましょう。

(1) 4×5 は、 6×3 より $\square$ 大きい。

(2) 7×3 は、 3×9 より 6 $\square$ 。

(3) どんな数に 0 をかけても答えは $\square$ です。また、0 にどんな数をかけても答えは $\square$ です。

3. $\square$ にあてはまる数を見つけましょう。

(1) $4 \times \square = 36$

(3) $3 \times \square = 27$

(5) $\square \times 9 = 36$

(7) $\square \times 1 = 0$

(9) $\square \times 7 = 63$

(2) $\square \times 8 = 24$

(4) $\square \times 2 = 18$

(6) $\square \times 7 = 21$

(8) $6 \times \square = 42$

(10) $8 \times \square = 32$

図 3-83 ケーススタディ（テスト・ドリル、単問）

この単元テストのコンテンツメタデータを、表 3-59 に示す。

表 3-59 ケーススタディのコンテンツメタデータ（テスト・ドリル）

項目名	最小	最大	値
テストメタ ID	1	1	http://smartschool.example.co.jp/P3/P030/U/003
テスト名	1	1	3 年「九九をおぼえよう」
テスト種別	1	1	単元テスト
学年	1	1	P3（小学第 3 学年）
教科	1	1	P030（算数（小学））
単元 ID	0	n	－
学習要素 ID	0	n	－
学習指導要領 ID	0	n	－
学習観点 ID	0	n	P030-101（算数への関心・意欲・態度） P030-102（数学的な考え方）
予定実施回数	0	1	1
満点	1	1	22
満点（学習観点別）	0	n	P030-101（算数への関心・意欲・態度）→4 P030-102（数学的な考え方）→18
問題メタ ID	0	n	http://smartschool.example.co.jp/P3/P030/U/003/1/1 （残りの問題は省略）
下位テストメタ ID	0	n	－

「問題メタ ID」に設定されているのは、図 3-83 の赤枠で囲った単問を指している（便宜上、残りの問題については省略した）。この赤枠で囲った単問のコンテンツメタデータを表 3-60 に示す。

表 3-60 ケーススタディのコンテンツメタデータ（単問）

項目名	最小	最大	値
単問メタ ID	1	1	http://.sample.co.jp/P3/P030/U/003/1/1
単問名	0	1	九九をおぼえよう 問 1 の (1)
問題	0	1	http://smartschool.example.co.jp/P3/P030/U/003/1/1/q
学年	1	1	P3（小学第 3 学年）
教科	1	1	P030（算数（小学））
単元 ID	0	1	－
学習要素 ID	0	n	－
学習指導要領 ID	0	n	－
学習観点 ID	0	n	P030-101（算数への関心・意欲・態度）
正答（模範解答）	0	n	http://smartschool.example.co.jp/P3/P030/U/003/1/1/c
誤答（誤答例）	0	n	http://smartschool.example.co.jp/P3/P030/U/003/1/01/w
配点	1	1	1
難易度	0	1	1

本ケーススタディで用いる上記、テスト・ドリル、単問の解答結果を、図 3-84、表 3-61、表 3-62 に示す。

スマスクドリル① きほん

学習日 2019年 12月 1日

単元	3年 1組 10番	16 22問
3年「九九をおぼえよう」	氏名 そうむ たろう	

1. 次の計算をしましょう。

(1) $2 \times 9 = 18$ (2) $5 \times 3 = 15$

(3) $8 \times 7 = 58$ (4) $4 \times 9 = 36$

(5) $3 \times 2 = 5$ (6) $2 \times 7 = 15$

(7) $0 \times 1 = 0$ (8) $4 \times 0 = 0$

2. にあてはまる数や言葉を書きましょう。

(1) 4×5 は、 6×3 より 大きい。

(2) 7×3 は、 3×9 より 小さい。

(3) どんな数に0をかけても答えは です。また、0にどんな数をかけても答えは です。

3. にあてはまる数を見つけましょう。

(1) $4 \times \text{ } = 36$ (2) $\text{ } \times 8 = 24$

(3) $3 \times \text{ } = 27$ (4) $\text{ } \times 2 = 18$

(5) $\text{ } \times 9 = 36$ (6) $\text{ } \times 7 = 21$

(7) $\text{ } \times 1 = 0$ (8) $6 \times \text{ } = 42$

(9) $\text{ } \times 7 = 63$ (10) $8 \times \text{ } = 32$

図 3-84 ケーススタディ（テスト・ドリル、単問の解答結果）

表 3-61 ケーススタディの結果データ（テスト・ドリル）

項目名	最小	最大	値
テスト結果 ID	1	1	F94DEB1A-C924-492E-A853-7E45BABF0FAC
テストメタ ID	1	1	http://smartschool.example.co.jp/P3/P030/U/003
完了日時	1	1	2018-04-12 10:30:00
学校生活場面	0	1	3rd_period（3 時間目）
実施回数	0	1	1
得点	1	1	16
得点（観点別）	0	n	P030-101（算数への関心・意欲・態度）→2 P030-102（数学的な考え方）→14
システム連携用児童生徒共通 ID	1	1	S0021756

表 3-62 ケーススタディの結果データ（単問）

項目名	最小	最大	値
単問結果 ID	1	1	E11EB4F7-B2D4-4006-8FED-9DD6C47285AA
単問メタ ID	1	1	http://smartschool.example.co.jp/P3/P030/U/003/01/01
完了日時	1	1	2018-04-12 10:15:00
学校生活場面	0	n	3rd_period（3 時間目）
実施回数	0	1	1
解答	0	1	18
正誤	1	1	True
得点	1	1	1
児童生徒 ID	1	1	S0021756

以下、テスト・ドリル、単問の解答結果を取得する場合の、リクエストおよびレスポンスを見てみる。

(1) テスト・ドリルの解答結果

① リクエスト

上記のテスト・ドリルの解答結果を取得する場合のリクエストパラメータを図 3-85 に示す。

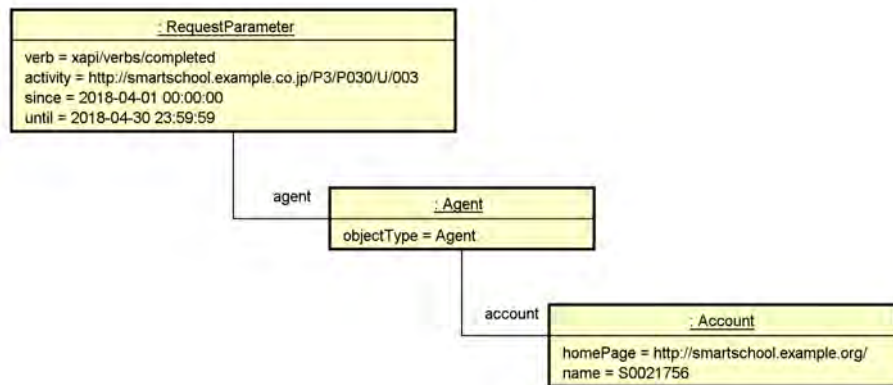


図 3-85 リクエストパラメータ（テスト・ドリルの解答結果）

ここでは、児童生徒 ID に「S0021756」（畠山 莉緒）、テスト・ドリルに「http://smartschool.example.co.jp/P3/P030/U/003」（3 年「九九の表とかけ算」）、期間に「2018-04-01～2018-04-30」を指定している。

また、Verb に「xapi/verbs/completed」（完了）を指定することにより、テスト・ドリルの解答完了のステートメントを取得するようフィルタリングをかけている。

図 3-85 を URL パラメータで記述すると以下ようになる（見やすくするために途中、改行して表示している）。

```
?agent={"objectType":"Agent","account":{"homePage":"http://smartschool.example.org/","name":"S0021756"}}
&verb=https://smartschool.example.org/xapi/verbs/completed
&activity=http://smartschool.example.co.jp/P3/P030/U/003
&since=2018-04-01%2000:00:00
&until=2018-04-30%2023:59:59
```

② レスポンス

テスト・ドリルの解答結果を問い合わせたレスポンス（表 3-61 にある結果データ）を図 3-86 に示す。

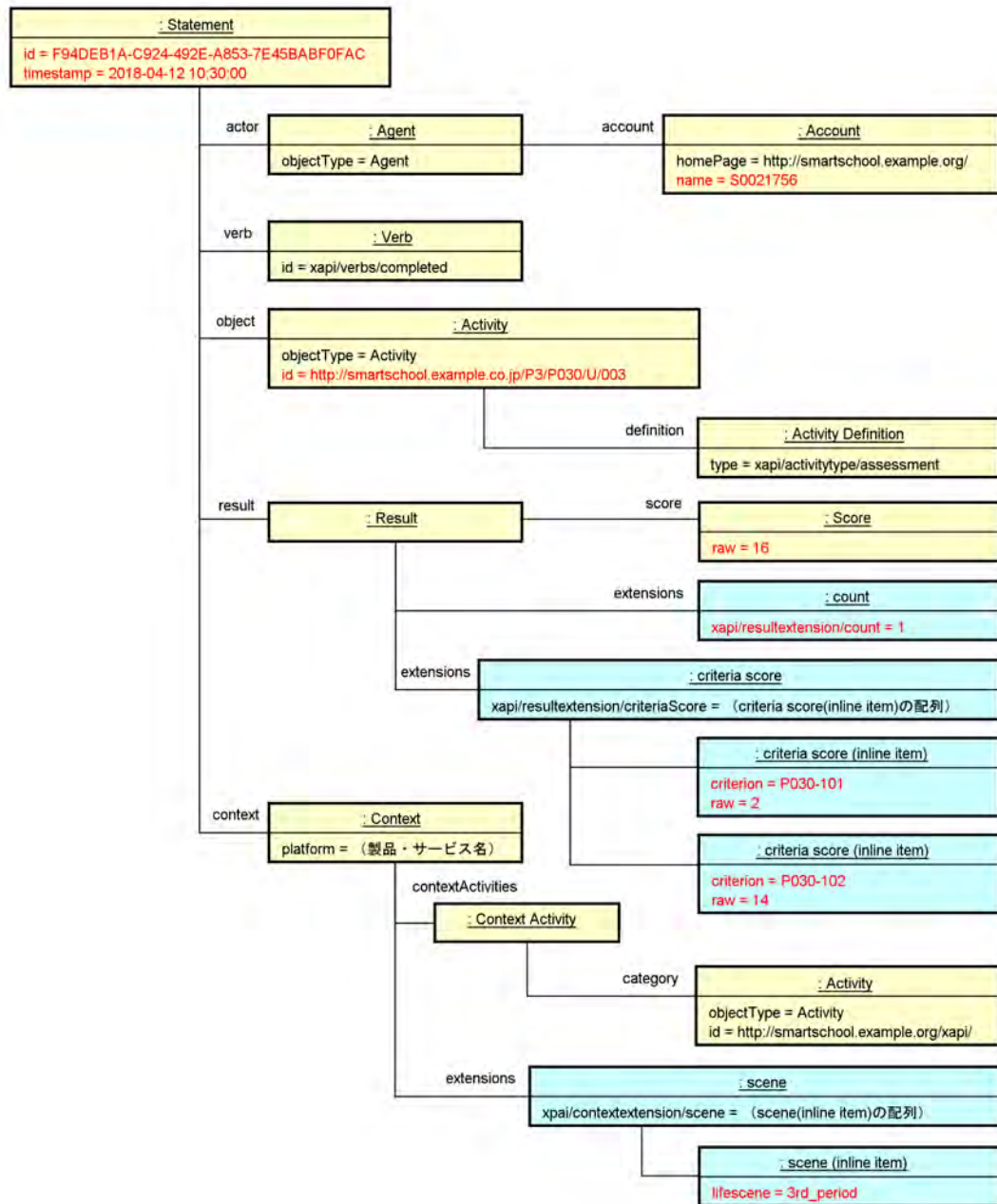


図 3-86 レスポンス（テスト・ドリルの解答結果）

図 3-86 を JSON 形式で表現すると以下ようになる。

```
{
  "timestamp": "2018-04-12T10:30:00+09:00",
  "id": "F94DEB1A-C924-492E-A853-7E45BABF0FAC",
  "actor": {
    "account": {
      "name": "S0021756",
      "homePage": "http://smartschool.example.org"
    },
    "objectType": "Agent"
  },
  "verb": {
    "id": "xapi/verbs/completed"
  },
  "object": {
    "id": "http://smartschool.example.co.jp/P3/P030/U/003",
    "definition": {
      "type": "xapi/activitytype/assessment"
    },
    "objectType": "Activity"
  },
  "result": {
    "score": {
      "raw": 16
    },
    "extensions": [
      {
        "xapi/resulttextension/count": "1"
      },
      {
        "xapi/resulttextension/criteriaScore": [
          {
            "criterion": "P030-101",
            "raw": "2"
          }
        ]
      }
    ]
  }
}
```

```

        {
            "criterion": "P030-102",
            "raw": "14"
        }
    ]
}
],
"context": {
    "contextActivities": {
        "category": [
            {
                "objectType": "Activity",
                "id": "http://smartschool.example.org/xapi/"
            }
        ]
    },
    "platform": "（製品・サービス名）",
    "extensions": {
        "xapi/contextextension/scene": [
            {
                "lifescene": "3rd_period"
            }
        ]
    }
}
}

```

(2) 単問の解答結果

① リクエスト

次に単問の解答結果の場合のリクエストパラメータを図 3-87 に示す。

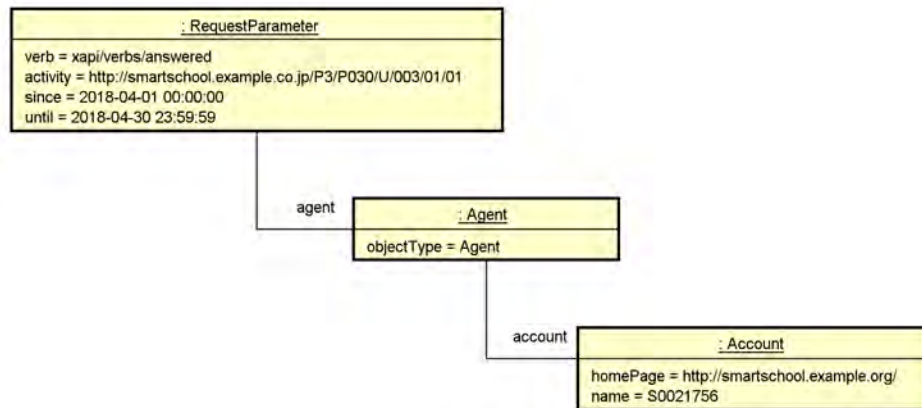


図 3-87 リクエストパラメータ（単問の解答結果）

児童生徒 ID に「S0021756」（総務 太郎）、単問に「http://smartschool.example.co.jp/P3/P030/U/003/1/1」（九九の表とかけ算 問1の(1)）、期間に「2018-04-01～2018-04-30」を指定している。

また、Verb に「xapi/verbs/answered」（解答・回答）を指定することにより、単問の解答完了のステートメントを取得するようフィルタリングをかけている。

図 3-87 を URL パラメータで記述すると以下ようになる（見やすくするために途中、改行して表示している）。

```
?agent={"objectType":"Agent","account":{"homePage":"http://smartschool.example.org/","name":"S0021756"}}
&verb= https://smartschool.example.org/xapi/verbs/answered
&activity= http://smartschool.example.co.jp/P3/P030/U/003/1/1
&since=2018-04-01%2000:00:00
&until=2018-04-30%2023:59:59
```

② レスポンス

テスト・ドリルの解答結果を問い合わせたレスポンス（表 3-62 にある結果データ）を図 3-88 に示す。

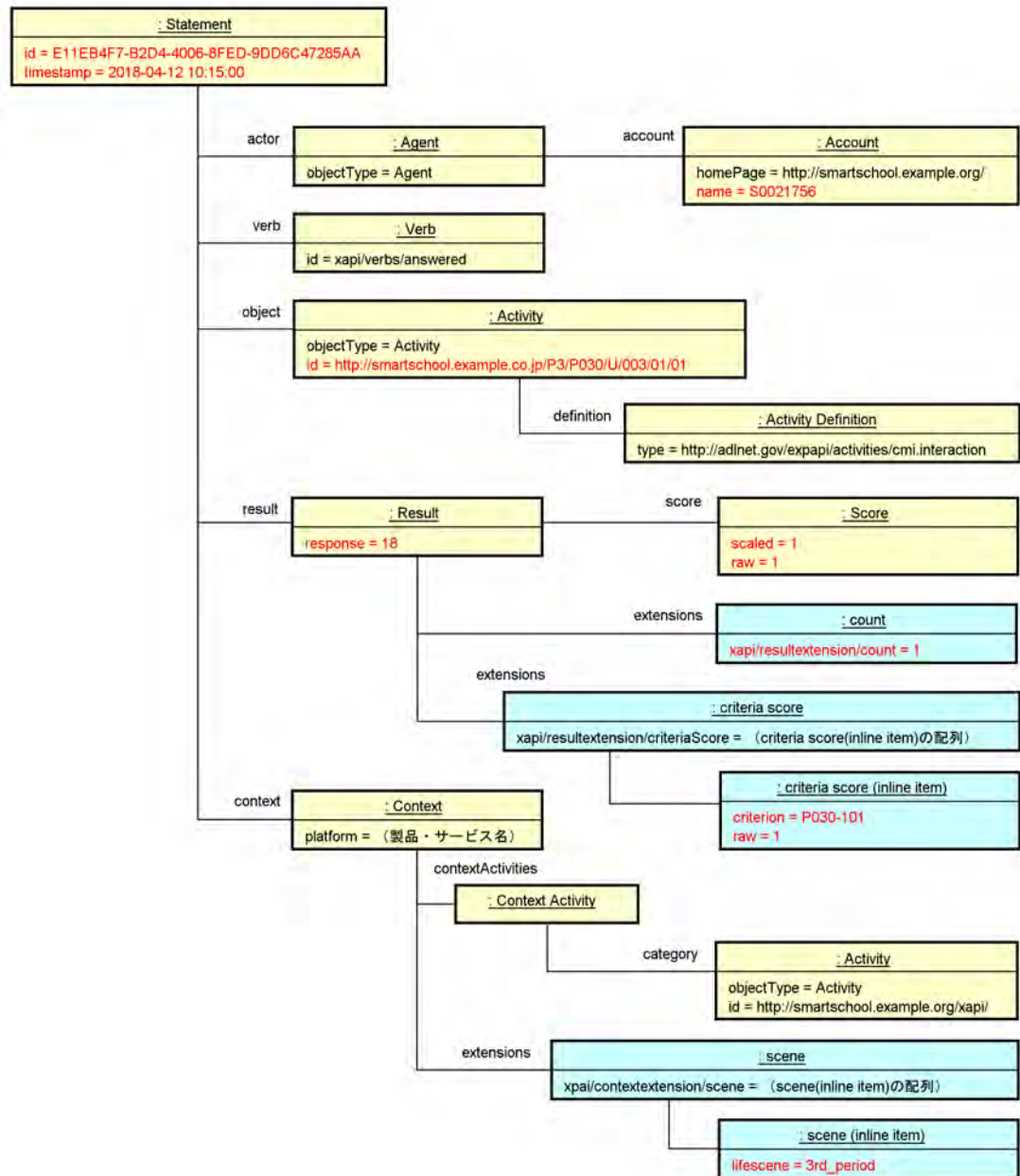


図 3-88 レスポンス（単問の解答結果）

図 3-88 を JSON 形式で表現すると以下のようなになる。

```
{
  "actor": {
    "account": {
      "name": " S0021756 ",
      "homePage": " http://smartschool.example.org/"
    },
    "objectType": "Agent"
  },
  "verb": {
    "id": "xapi/verbs/answered"
  },
  "object": {
    "id": "http://smartschool.example.co.jp/P3/P030/U/003/01/01",
    "definition": {
      "type": "http://adlnet.gov/expapi/activities/cmi.interaction"
    },
    "objectType": "Activity"
  },
  "result": {
    "score": {
      "scaled": 1,
      "raw": 1
    },
    "response": "18",
    "extensions": [
      {
        "xapi/resultextension/count": "1"
      },
      {
        "xapi/resultextension/criteriaScore": [
          {
            "criterion": "P030-101",
            "raw": "1"
          }
        ]
      }
    ]
  }
}
```

```

        ]
    }
]
},
"context": {
    "contextActivities": {
        "category": [
            {
                "id": "http://smartschool.example.org/xapi/",
                "objectType": "Activity"
            }
        ]
    },
    "platform": "（製品・サービス名）",
    "extensions": {
        "xapi/contextextension/scene": [
            {
                "lifescene": "3rd_period"
            }
        ]
    }
}
}

```

3.10.5.2 アンケート設問の回答結果を連携する

ここではアンケートの例として、図 3-89 にあるような小学第 3 学年算数の単元テストを用いる。

学習に関するアンケート

第 回 (月)

年 組 番

氏名

①あてはまる ②ややあてはまる ③あまりあてはまらない ④あてはまらない

		①	②	③	④
1	授業で分からないことがあると、先生に聞くことができる。				
2	もっとうまい解き方や別の考え方はないかと考える。				
3	良く分からないことは、分かるまで調べたい。				
4	興味のあることは調べずにはいられない。				
5	先生は学習のことについてほめてくれる。				
6	自分は勉強がよくできると思う。				
7	学校では落ち着いて授業を受けている				
8	難しい問題にであうと、よりやる気がでる。				
9	自分から勉強に取り組んでいる。				
10	分からないことがあると、いろいろな方法で調べている。				
11	授業では友達と協力して学ぶことが多い。				
12	今までよりも、難しい問題に取り組むことが多い。				
13	学校では落ち着いて授業を受けている。				
14	分からないことがあったら、先生に質問をする。				
15	テストがあれば、自分で計画を立てて勉強する。				
16	授業では友達に教えたり、教わったりすることが多い。				
17	できるだけ自分ひとりの力で解決しようとする。				
18	失敗しても学ぶことは面白い。				
19	クラスは発言しやすい雰囲気である。				
20	いろいろなことを学ぶのは面白い。				

図 3-89 ケーススタディ（アンケート）

このアンケートのコンテンツメタデータを、表 3-63 に示す。

表 3-63 ケーススタディのコンテンツメタデータ（アンケート）

項目名	最小	最大	値
アンケートメタ ID	1	1	http://smartschool.example.co.jp/P1/E
アンケート名	1	1	学習に関するアンケート
アンケート種別	1	1	(学習に関するアンケート)
学年	1	1	P6 (小学第 6 学年)
教科	1	1	n/a (該当なし)
単元 ID	0	1	－
学習要素 ID	0	n	－
学習指導要領 ID	0	n	－
学習観点 ID	0	n	－
予定実施回数	0	1	1
設問メタ ID	0	n	http://smartschool.example.co.jp/P1/E/001/1 (残りの設問は省略)
下位アンケートメタ ID	0	n	－

「設問メタ ID」に設定されているのは、図 3-89 の赤枠で囲った単問を指している（便宜上、残りの問題については省略した）。この赤枠で囲った設問のコンテンツメタデータを表 3-64 に示す。

表 3-64 ケーススタディのコンテンツメタデータ（設問）

項目名	最小	最大	値
設問メタ ID	1	1	http://smartschool.example.co.jp/P1/E/001/1
設問名	0	1	学習に関するアンケート 設問 1
設問	0	1	http://smartschool.example.co.jp/P1/E/001/1/q
学年	1	1	P6 (小学第 6 学年)
教科	1	1	n/a (該当なし)
単元 ID	0	1	－
学習要素 ID	0	n	－
学習指導要領 ID	0	n	－
学習観点 ID	0	n	－

本ケーススタディで用いる上記、アンケート設問の回答結果を、図 3-90、表 3-65 に示す。

学習に関するアンケート

第 1 回 (9 月)

6 年	3 組	10 番
氏名		
総務 花子		

①あてはまる ②ややあてはまる ③あまりあてはまらない ④あてはまらない

	①	②	③	④
1 授業で分からないことがあると、先生に聞くことができる。	☒			
2 もっとうまい解き方や別の考え方はないかと考える。	☐			
3 良く分からないことは、分かるまで調べたい。	☐			
4 興味のあることは調べずにはいられない。		☐		
5 先生は学習のことについてほめてくれる。		☐		
6 自分は勉強がよくできと思う。			☐	
7 学校では落ち着いて授業を受けている		☐		
8 難しい問題にであうと、よりやる気がでる。		☐		
9 自分から勉強に取り組んでいる。	☐			
10 分からないことがあると、いろいろな方法で調べている。	☐			
11 授業では友達と協力して学ぶことが多い。		☐		
12 今までよりも、難しい問題に取り組むことが多い。			☐	
13 学校では落ち着いて授業を受けている。		☐		
14 分からないことがあったら、先生に質問をする。		☐		
15 テストがあれば、自分で計画を立てて勉強する。				☐
16 授業では友達に教えたり、教わったりすることが多い。				☐
17 できるだけ自分ひとりの力で解決しようとする。	☐			
18 失敗しても学ぶことは面白い。		☐		
19 クラスは発言しやすい雰囲気である。	☐			
20 いろいろなことを学ぶのは面白い。	☐			

図 3-90 ケーススタディ（アンケートの回答結果）

表 3-65 ケーススタディの結果データ（設問）

項目名	最小	最大	値
設問結果 ID	1	1	D3D448BC-C4AA-4906-A17C-A19FC6EAFE55
設問メタ ID	1	1	http://smartschool.example.co.jp/P1/E/001/1
完了日時	1	1	2018-09-03 20:15:00
学校生活場面	0	n	inhome（自宅）
実施回数	0	1	1
回答	0	1	1
児童生徒 ID	1	1	S0069868

以下、アンケート設問の回答結果を取得する場合の、リクエストおよびレスポンスを見てみる。

(1) アンケート設問の回答結果

① リクエスト

上記のアンケート設問の回答結果を取得する場合のリクエストパラメータを図 3-91 に示す。

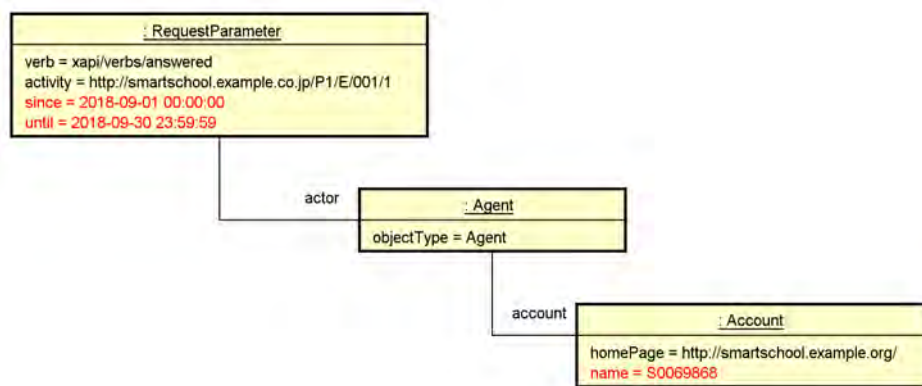


図 3-91 リクエストパラメータ（アンケート設問の回答結果）

ここでは、児童生徒 ID に「S0069868」（総務 花子）、テスト・ドリルに「http://smartschool.example.co.jp/P1/E/001/1」（学習に関するアンケート 設問 1）、期間に「2018-09-01～2018-09-30」を指定している。

また、Verb に「xapi/verbs/answered」（解答・回答）を指定することにより、設問の回答完了のステートメントを取得するようフィルタリングをかけている。

図 3-91 を URL パラメータで表現すると以下のようなになる（見やすくするために途中、改行して表示している）。

```
?agent={"objectType":"Agent","account":{"homePage":"http://smartschool.example.org/","name":"S0069868"}}
&verb= https://smartschool.example.org/xapi/verbs/answered
&activity= http://smartschool.example.co.jp/P1/E/001/1
&since=2018-09-01%2000:00:00
&until=2018-09-30%2023:59:59
```

② レスポンス

アンケート設問の回答結果を問い合わせたレスポンス（表 3-65 にある結果データ）を図 3-92 に示す。

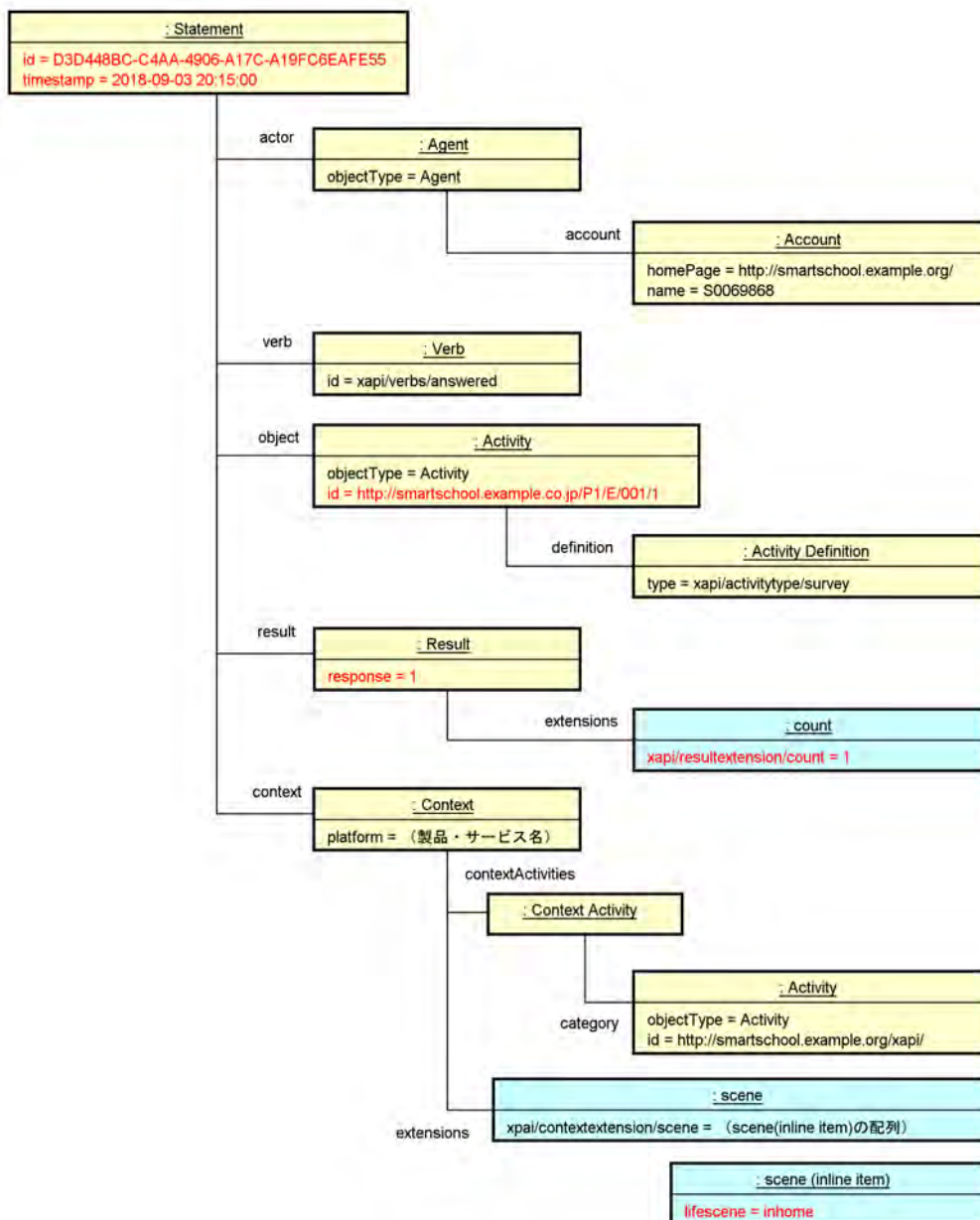


図 3-92 レスポンス（アンケート設問の回答結果）

図 3-92 を JSON 形式で表現すると以下のようなになる。

```
{
  "timestamp": "2018-05-09T20:15:00+09:00",
  "id": " D3D448BC-C4AA-4906-A17C-A19FC6EAFE55",
  "actor": {
    "account": {
      "name": "S0069868",
      "homePage": "http://smartschool.example.org"
    },
    "objectType": "Agent"
  },
  "verb": {
    "id": "xapi/verbs/answered"
  },
  "object": {
    "id": " http://smartshool.example.co.jp/P1/E/001/1 ",
    "definition": {
      "type": "xapi/activitytype/survey"
    },
    "objectType": "Activity"
  },
  "result": {
    "response": "1",
    "extensions": [
      {
        "xapi/resultextension/count": "1"
      },
    ]
  },
  "context": {
    "contextActivities": {
      "category": [
        {
          "id": "http://smartschool.example.org/xapi/",
          "objectType": "Activity"
        }
      ]
    }
  }
}
```

```

        }
    ]
},
"platform": "（製品・サービス名）",
"extensions": {
    "xapi/contextextension/scene": [
        {
            "lifescene": "inhome"
        }
    ]
}
}
}

```

4. API サービスの利用

本章では、スマートスクール・プラットフォームにおけるシステム間データ連携およびデータ活用事例について解説する。

4.1 API サービスによるシステム間データ連携

スマートスクール・プラットフォーム技術仕様 API によりシステム連携をする際は、各々の API サービスを個別に用いるのではなく、いくつかの API を組み合わせることにより、一連のデータ連携を実現することになる。

ここでは、システム導入時、年次処理、日次処理のそれぞれの場面ごとに、具体的なシステム間データ連携の手順について解説する。表 4-1 に、データ連携する場面と、連携対象となるデータを示す。

表 4-1 データ連携契機と連携対象データ

データ連携する場面	連携対象となるデータ										
	児童生徒情報	教職員情報	学級情報	異動情報	時間割情報	出欠情報	日常所見情報	保健室利用情報	テスト結果情報	ドリル結果情報	アンケート結果情報
(システム導入時)											
システムセットアップ	●	●	●		●						
(年次処理)											
児童生徒の入学・卒業等 学級編成 時間割作成	●	●	●		●						
(日次処理)											
時間割変更					●						
児童生徒の転学等	●			●							
教職員の異動		●									

データ連携する場面	連携対象となるデータ										
	児童生徒情報	教職員情報	学級情報	異動情報	時間割情報	出欠情報	日常所見情報	保健室利用情報	テスト結果情報	ドリル結果情報	アンケート結果情報
児童生徒の生活状況の把握						●	●	●			
児童生徒の学習理解度や自己認識状況の把握									●	●	●

4.1.1 システム導入時

4.1.1.1 システムセットアップ

授業・学習系システムの初期導入時に、校務系システムに登録されている児童生徒情報、教職員情報、学級情報を、授業・学習系システムへ登録する手続きをスマートスクール・プラットフォーム技術仕様 API で行う。ここでは例として、連携対象となる児童生徒情報、教職員情報、学級情報が、校務系システムに登録されていると想定し、それらを授業・学習系システムに連携する場合について解説する。システム初期導入時には、登録するデータが大量になることから、API を用いないデータ連携（ファイル連携等）が一般的であるが、校務系システムがスマートスクール・プラットフォーム技術仕様に準拠している場合、API を用いたデータ連携を用いた方法も有効になる。

図 4-1 にシステムセットアップ時の連携処理を示す。

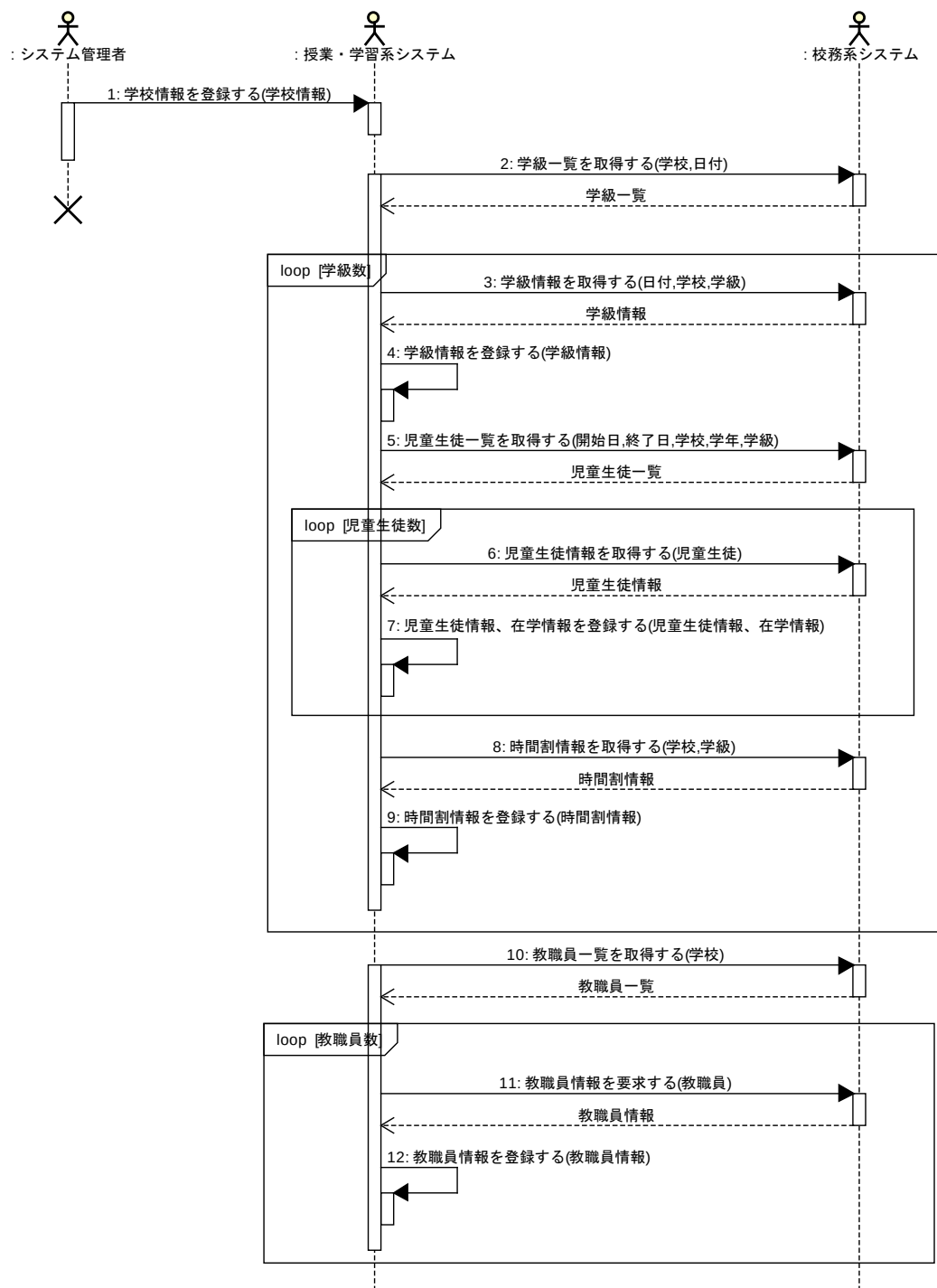


図 4-1 システム連携処理（システムセットアップ）

以下、システム連携の具体的な手続きを説明する。

1. システム管理者は、事前に授業・学習系システムの入力画面等を通して、授業・学習系システムに学校情報を登録する。学校情報については、スクール・プラットフォーム技術仕様 API による連携対象ではないため、授業・学習系システムに、事前に登録しておくことになる。
2. 授業・学習系システムは、校務系システムの学級情報 API サービスを呼び出し、当該学校の年度初日（4月1日）時点の学級の一覧を取得する。
3. 授業・学習系システムは、2で取得した学級一覧より、校務系システムの学級情報 API サービスを呼び出し、各学級の学級情報を取得する。
4. 授業・学習系システムは、3で取得した学級情報を自システムに登録する。
5. 授業・学習系システムは、3で取得した学級情報から、校務系システムの児童生徒情報 API サービスを呼び出し、当該学級に所属する児童生徒の一覧を取得する。
6. 授業・学習系システムは、5で取得した児童生徒一覧より、校務系システムから児童生徒情報 API サービスを呼び出し、各児童生徒の児童生徒情報を取得する。
7. 授業・学習系システムは、6で取得した児童生徒情報を自システムに登録する。
8. 授業・学習系システムは、3で取得した学級情報から、校務系システムの時間割情報 API サービスを呼び出し、当該学級の時間割情報を取得する。
9. 授業・学習系システムは、8で取得した時間割情報を自システムに登録する。
10. 授業・学習系システムは、校務系システムの教職員情報 API サービスを呼び出し、当該学校に所属する教職員の一覧を取得する。
11. 授業・学習系システムは、10で取得した教職員一覧より、校務系システムの教職員情報 API サービスを呼び出し、各教職員の教職員情報を取得する。
12. 授業・学習系システムは、11で取得した教職員情報を自システムに登録する。

4.1.2 年次処理

4.1.2.1 児童生徒の入学・卒業等、学級編成、時間割作成

新年度の児童生徒の入学、進級、卒業に伴う学級編成、時間割作成時に、校務系システムに登録された学級情報、児童生徒情報、在学情報、時間割情報、教職員情報を、授業・学習系システムに連携する。基本的には、年次処理と同様の処理となる。

ここでは例として、連携対象となる学級情報、児童生徒情報、在学情報、時間割情報、教職員情報が、校務系システムに登録されていると想定し、それらを授業・学習系システムに連携する場合について解説する。

図 4-2 に年次処理におけるシステム連携処理を示す。

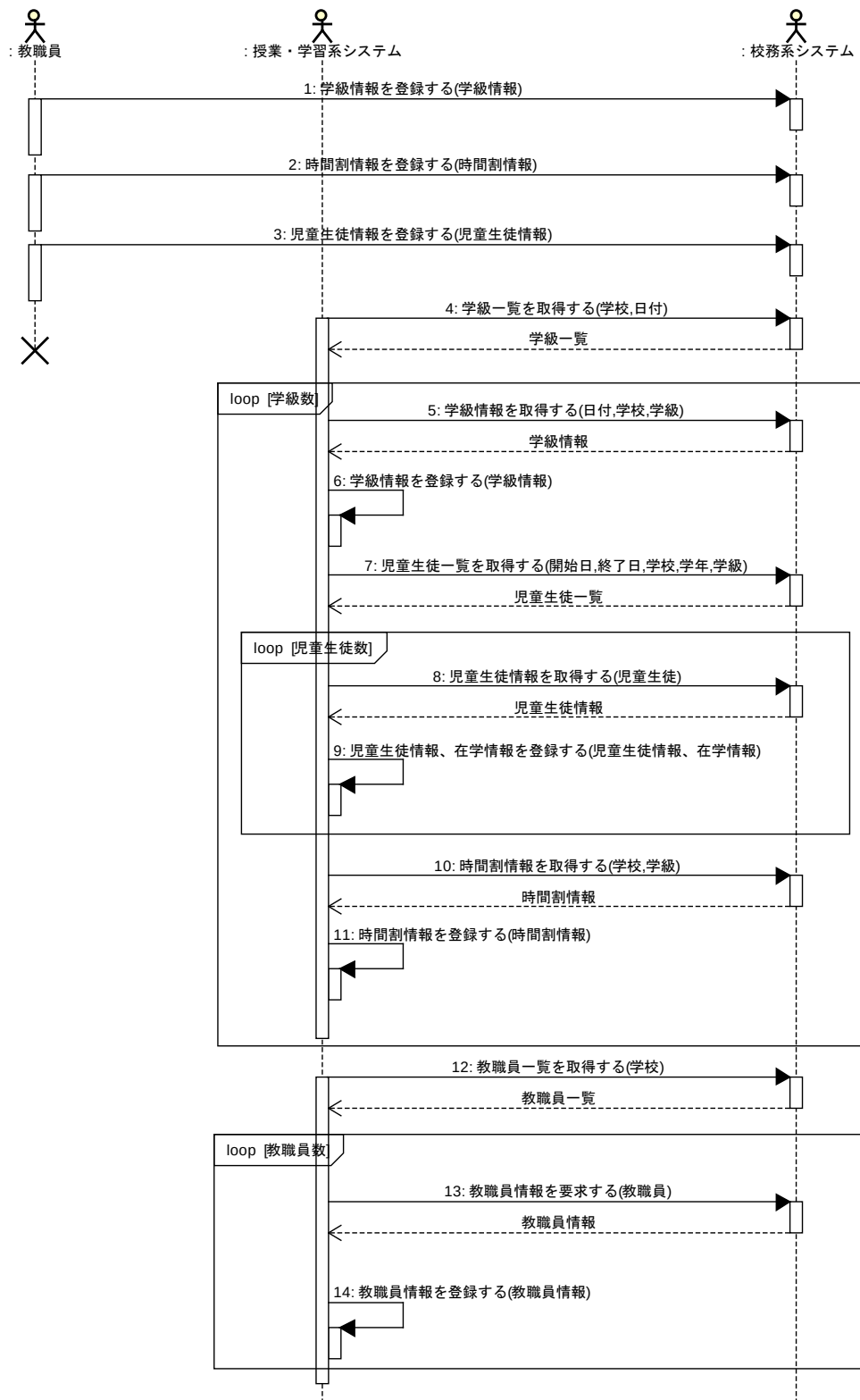


図 4-2 システム連携処理（児童生徒の入学・卒業等、学級編成、時間割作成）

以下、システム連携の具体的な手続きを説明する。

1. 教職員は、新年度時に学級編成を行ったら、校務系システムに学級情報を登録する。
2. 教職員は、新年度時に時間割編成を行ったら、校務系システムに時間割情報を登録する。
3. 教職員は、新年度時に校務系システムに、新入学生の児童生徒情報を登録する。
また、児童生徒の進級、卒業に伴い当該児童生徒の在学情報を更新する。
4. 授業・学習系システムは、校務系システムの学級情報 API サービスを呼び出し、当該学校の年度初日（4月1日）時点の学級の一覧を取得する。
5. 授業・学習系システムは、2で取得した学級一覧より、校務系システムの学級情報 API サービスを呼び出し、各学級の学級情報を取得する。
6. 授業・学習系システムは、3で取得した学級情報を自システムに登録する。
7. 授業・学習系システムは、3で取得した学級情報から、校務系システムの児童生徒情報 API サービスを呼び出し、当該学級に所属する児童生徒の一覧を取得する。
8. 授業・学習系システムは、5で取得した児童生徒一覧より、校務系システムから児童生徒情報 API サービスを呼び出し、各児童生徒の児童生徒情報を取得する。
9. 授業・学習系システムは、6で取得した児童生徒情報を自システムに登録する。
10. 授業・学習系システムは、3で取得した学級情報から、校務系システムの時間割情報 API サービスを呼び出し、当該学級の時間割情報を取得する。
11. 授業・学習系システムは、8で取得した時間割情報を自システムに登録する。
12. 授業・学習系システムは、校務系システムの教職員情報 API サービスを呼び出し、当該学校に所属する教職員の一覧を取得する。
13. 授業・学習系システムは、10で取得した教職員一覧より、校務系システムの教職員情報 API サービスを呼び出し、各教職員の教職員情報を取得する。
14. 授業・学習系システムは、11で取得した教職員情報を自システムに登録する。

4.1.3 日次処理

日次処理におけるシステム連携として、以下を想定している。

- 時間割変更
- 児童生徒の転学等
- 教職員の異動
- 児童生徒の生活状況の把握
- 児童生徒の学習理解度や自己認識状況の把握

上記のイベントは毎日発生するものではなく、実際にデータが連携されるのは、当該イベントが発生した都度というのが正確であるが、スマートスクール・プラットフォーム技術仕様 API は、データを取得する側のシステムが、データの取得元のシステムに問い合わせる仕組みになっているため、データを取得する側のシステムから定期的にシステム連携処理を実行する必要がある。

また、上記の日次処理について、スマートスクール・プラットフォーム技術仕様 API では、取得時点での全データを返す仕様となっており、すでに取得済みのデータとの差分だけを連携する仕様にはなっていない。そのため、すべての連携処理において、取得する側のシステムでは、連携処理の都度、全データを取得して登録、更新、削除する必要がある点に留意する必要がある。

以下、それぞれの処理の詳細について解説する。

4.1.3.1 時間割変更

担当教諭が、学習指導計画に基づき時間割情報を変更した場合に、日次処理において、時間割情報を授業・学習系システムに連携する。ここでは例として、時間割情報が、校務系システムに登録されていると想定し、それを授業・学習系システムに連携する場合について解説する。

図 4-3 に時間割変更時のシステム連携処理を示す。

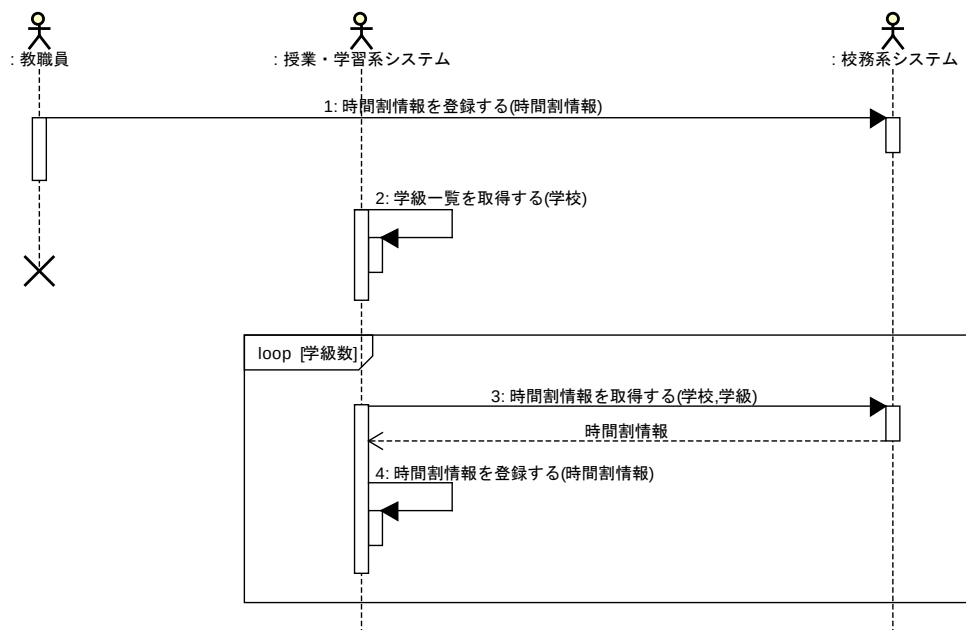


図 4-3 システム連携処理（時間割変更）

以下、時間割変更時の具体的なシステム連携の処理内容を説明する。

1. 教職員は、時間割を変更したら、校務系システムに時間割情報を登録する。
2. 授業・学習系システムは、自システムに登録されている学級一覧を取得する。
3. 授業・学習系システムは、2で取得した学級一覧より、学校、学級を指定し、校務系システムの時間割情報 API サービスを呼び出し、各学級の時間割情報を取得する。
4. 授業・学習系システムは、3で取得した時間割情報を自システムに登録する。

4.1.3.2 児童生徒の転学等

児童生徒の転入学、編入学、転学、退学、去校時に、日次処理において、児童生徒情報、異動情報を校務系システムから授業・学習系システムに連携する。システムセットアップ時、年次処理においても、児童生徒情報を連携するが、児童生徒の異動情報についても連携する点が異なる。

図 4-4 に児童生徒の転学等時のシステム連携処理を示す。

児童生徒の転学等に伴うシステム連携処理は 2 つの方法がある。

ひとつは、児童生徒一人ずつ処理を行う方法で、最初に対象学校の学級ごとに、当該学級に所属する児童生徒を取得し、児童生徒一人ずつ異動情報を取得して連携する方法である。異動情報を登録するまでの処理として、①学校に存在する学級一覧の取得→②学級に所属する児童生徒一覧の取得→③児童生徒一人ずつの異動情報の取得→④児童生徒の異動情報の登録、の順で処理を行う必要がある。

もうひとつの方法は、上記①、②、③の処理を省略して、学校、期間を指定すると、その条件に当てはまる児童生徒の異動情報の一覧を取得することができる。

上記 2 つの方法は、状況に応じて使い分ける必要がある。

また、ここでは、児童生徒の異動情報が、校務系システムに登録されていると想定し、それを授業・学習系システムに連携する場合について解説する。

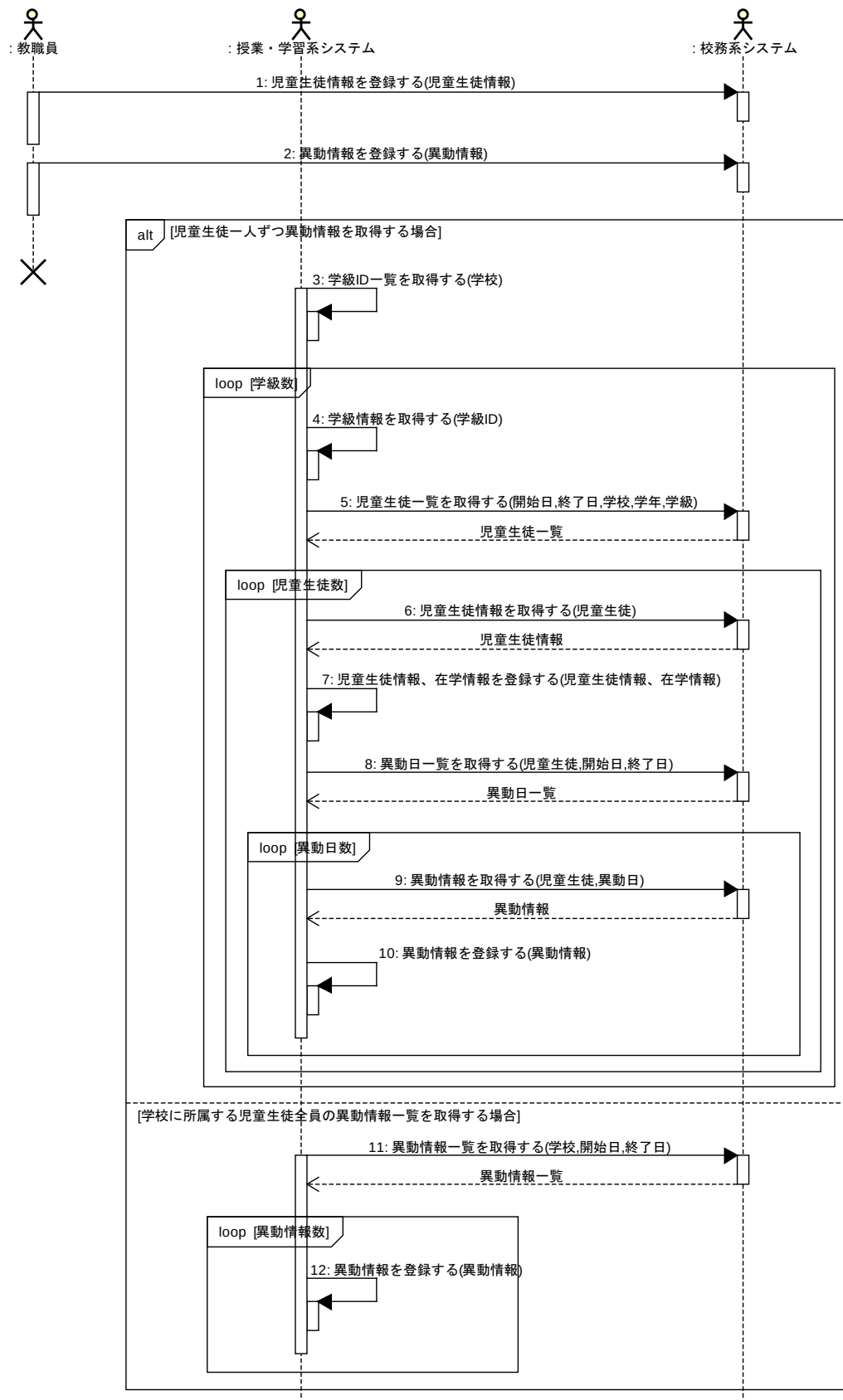


図 4-4 システム連携処理（児童生徒の転学等）

以下、児童生徒の転学時の具体的なシステム連携の処理内容について説明する。

(1) 児童生徒一人ずつ異動情報を取得する場合

1. 教職員は、児童生徒が転入学、編入学、転学、退学、去校したら、校務系システムの児童生徒情報、在学情報を登録する。
2. 1と同様に、教職員は、校務系システムに児童生徒の異動情報を登録する。
3. 授業・学習系システムは、自システムに登録されている学級の一覧を取得する。
4. 授業・学習系システムは、3で取得した学級一覧より、自システムに登録されている各学級の学級情報を取得する。
5. 授業・学習系システムは、4で取得した学級情報より、日付期間（例えば、本処理の実行日の前日日付）、学校、学年、学級を指定して、校務系システムの児童生徒情報 API サービスを呼び出し、当該学級に所属する児童生徒一覧を取得する。
6. 授業・学習系システムは、5で取得した児童生徒一覧より、児童生徒を指定して、校務系システムの児童生徒情報 API サービスを呼び出し、当該児童生徒の児童生徒情報を取得する。
7. 授業・学習系システムは、6で取得した児童生徒情報を自システムに登録する。
8. 授業・学習系システムは、5で取得した児童生徒一覧より、児童生徒、日付期間（例えば、本処理の前日日付）を指定して、校務系システムの異動情報 API サービスを呼び出し、当該児童生徒の異動日一覧を取得する。
9. 授業・学習系システムは、8で取得した異動日と、児童生徒を指定して、校務系システムの異動情報 API サービスを呼び出し、当該児童生徒の異動情報を取得する。
10. 授業・学習系システムは、9で取得した児童生徒の異動情報を自システムに登録する。

(2) 学校に所属する児童生徒全員の異動情報一覧を取得する場合

1. 教職員は、児童生徒が入学、転入学、編入学、転学、退学、去校、卒業したら、校務系システムの児童生徒情報、在学情報を登録する。
2. 1と同様に、教職員は、校務系システムに児童生徒の異動情報を登録する。
11. 授業・学習系システムは、学校、日付期間を指定して、校務系システムの異動情報 API サービスを呼び出し、当該学校、日付期間内に発生した児童生徒の異動情報一覧を取得する。
12. 授業・学習系システムは、11 で取得した児童生徒の異動情報一覧を自システムに登録する。

4.1.3.3 教職員の異動

教職員が、異動（転勤、退職、転属等）した場合や、休職等した場合に、日次処理において、ここでは例として、校務系システムに登録されている教職員情報を、授業・学習系システムに連携する場合について解説する。

図 4-5 に教職員の異動時のシステム連携処理を示す。

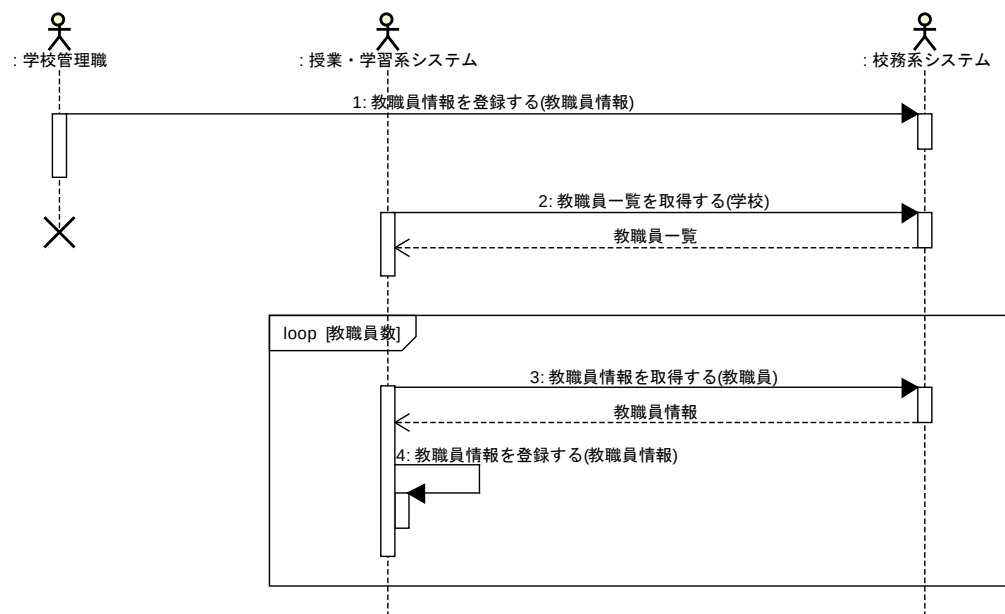


図 4-5 システム連携処理（教職員の異動）

以下、教職員の異動時のシステム連携の処理内容について説明する。

1. 学校管理職は、教職員が異動（転勤、退職、転属等）した場合や、休職等した場合に、当該教職員の教職員情報を校務系システムに登録する。
2. 授業・学習系システムは、学校を指定して、校務系システムの教職員情報 API サービスを呼び出し、当該学校の教職員一覧を取得する。
3. 授業・学習系システムは、2 で取得した教職員一覧より、教職員を指定して、校務系システムの教職員情報 API サービスを呼び出し、当該教職員の教職員情報を取得する。
4. 授業・学習系システムは、3 で取得した教職員情報を自システムに登録する。

4.1.3.4 児童生徒の生活状況等の把握

担当教諭、養護教諭が校務系システムに登録した日々の児童生徒の生活状況等（出欠情報、日常所見情報、保健室利用情報）を、日次処理にて、ここでは例として、校務系システムに登録されている出欠情報、日常所見情報、保健室利用情報を、授業・学習系システムに連携する場合について解説する。

図 4-6 に児童生徒の生活状況（出欠情報、日常所見情報、保健室利用情報）のシステム連携処理を示す。

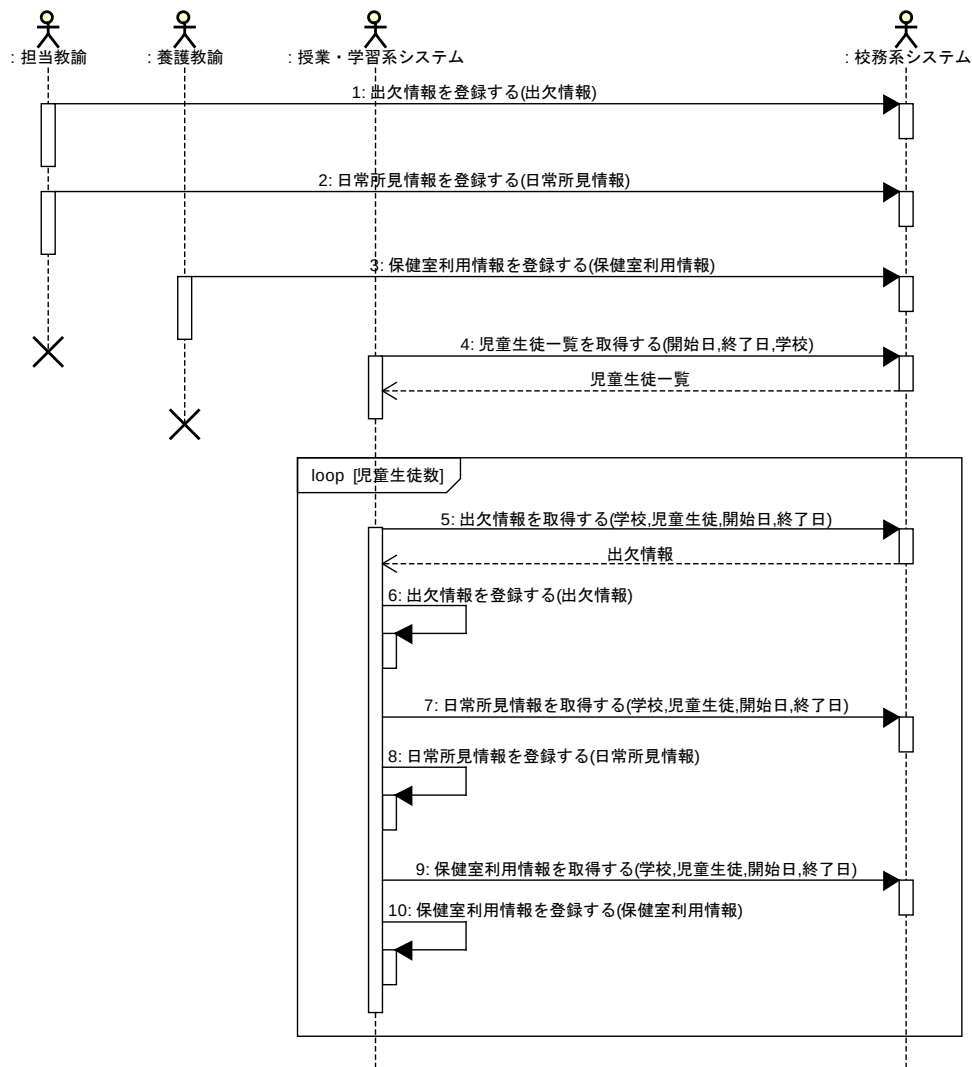


図 4-6 システム連携処理（児童生徒の生活状況の把握）

以下、児童生徒の生活状況等（出欠情報、日常所見情報、保健室利用情報）のシステム連携の処理内容を説明する。

1. 担当教諭は、校務系システムに児童生徒の日々の出欠情報を登録する。
2. 担当教諭は、校務系システムに児童生徒の日々の日常所見情報を登録する。
3. 養護教諭は、校務系システムに児童生徒の日々の保健室利用情報を登録する。
4. 授業・学習系システムは、日付期間（例えば、連携処理実行日の前日日付）、学校を指定して、校務系システムの児童生徒情報 API サービスを呼び出し、児童生徒一覧を取得する。
5. 授業・学習系システムは、4 で取得した児童生徒一覧より、学校、児童生徒、日付期間を指定して、校務系システムの出欠情報 API サービスを呼び出し、当該児童生徒の出欠情報を取得する。
6. 授業・学習系システムは、5 で取得した当該児童生徒の出欠情報を自システムに登録する。
7. 授業・学習系システムは、4 で取得した児童生徒一覧より、学校、児童生徒、日付期間を指定して、校務系システムの日常所見情報 API サービスを呼び出し、当該児童生徒の日常所見情報を取得する。
8. 授業・学習系システムは、7 で取得した当該児童生徒の日常所見情報を自システムに登録する。
9. 授業・学習系システムは、4 で取得した児童生徒一覧より、学校、児童生徒、日付期間を指定して、校務系システムの保健室利用情報 API サービスを呼び出し、当該児童生徒の保健室利用情報を取得する。
10. 授業・学習系システムは、9 で取得した当該児童生徒の保健室利用情報を自システムに登録する。

4.1.3.5 児童生徒の学習理解度や自己認識状況の把握

児童生徒の学習理解度や自己認識状況の把握では、ここでは例として、授業・学習システムに登録されている児童生徒のテスト・ドリル（定期テスト、単元テスト等）の解答結果や、アンケート設問の回答結果を、校務系システムに連携させることにより、当該児童生徒の学習指導や成績処理に活用する場合について解説する。

以下、テスト・ドリルの解答結果、アンケート設問の回答結果を連携させる場合の二つに分けて説明する。

(1) テスト・ドリルの解答結果を連携する場合

図 4-7 にテスト・ドリルの解答結果のシステム連携の処理内容を示す。

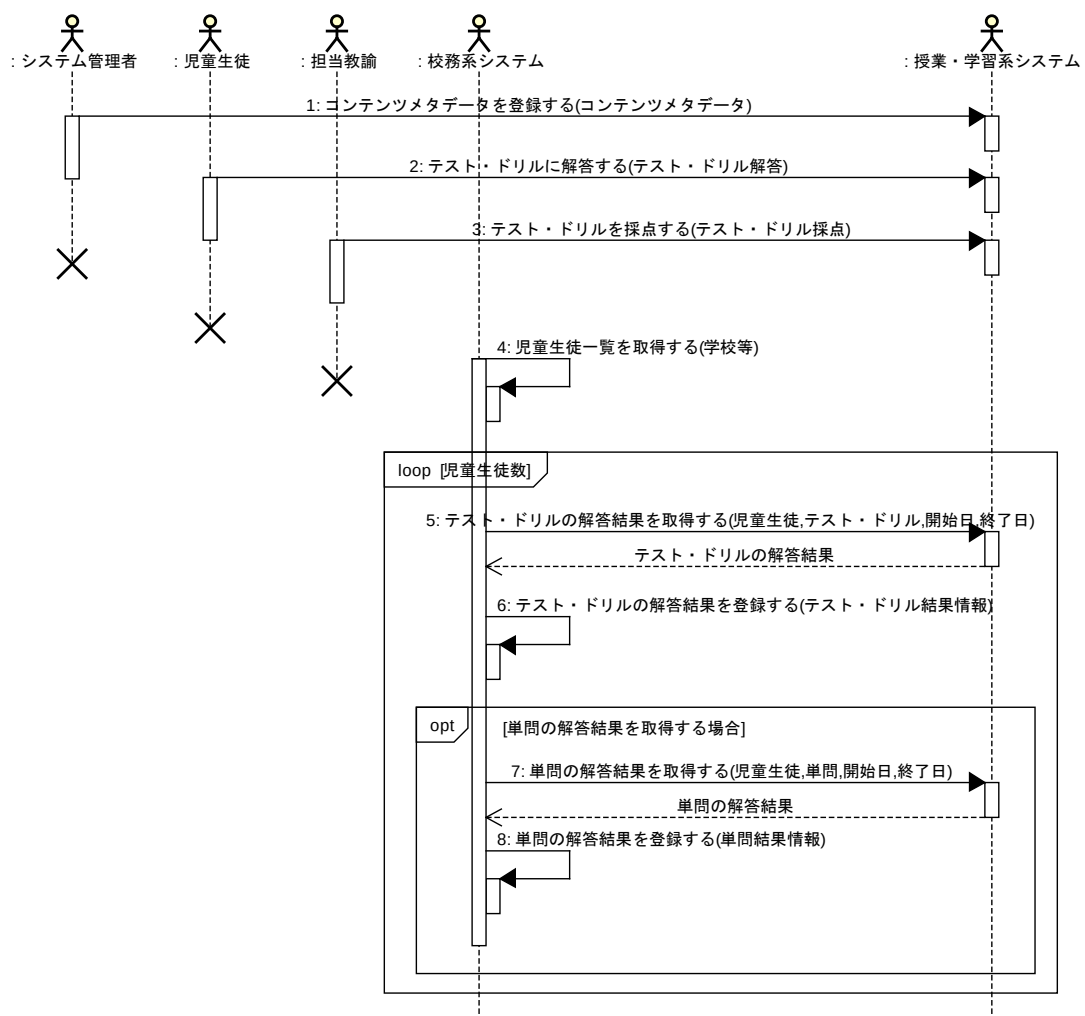


図 4-7 システム連携処理（児童生徒の学習理解度や自己認識状況の把握－テスト・ドリル）

以下、テスト・ドリルの解答結果のシステム連携の処理内容について説明する。

1. システム管理者は、授業・学習系システムにテスト・ドリルのコンテンツメタデータ²を登録する。コンテンツメタデータは、スマートスクール・プラットフォーム技術仕様 API の連携対象外のため、事前に登録しておく必要がある。
2. 児童生徒は、授業・学習系システムにて、テスト・ドリルに解答する。解答結果は授業・学習系システムに解答結果として登録される。
3. 担当教諭は、授業・学習系システムにて、児童生徒のテスト・ドリル解答結果を採点する。（ただし、通常、採点処理は授業・学習系システムにて自動で行うため、この処理は必要ない場合もある。）
4. 校務系システムは、自システムに登録されている児童生徒一覧を取得する。
5. 校務系システムは、4 で取得した児童生徒一覧より、各児童生徒のテスト・ドリルの解答結果を、児童生徒、対象のテスト・ドリル、日付期間を指定して、授業・学習系システムの学習記録情報 API サービスを呼び出し、取得する。
6. 校務系システムは、5 で取得した児童生徒のテスト・ドリルの解答結果を自システムに登録する。
7. 校務系システムは、必要な場合、テスト・ドリルの単問解答結果を、5 と同様に、授業・学習系システムの学習記録情報 API サービスを呼び出し、取得する。
8. 校務系システムは、7 で取得した児童生徒の単問の解答結果を自システムに登録する。

² コンテンツメタデータとは、テスト・ドリルの問題内容、配点、満点等のことである。それに対して、児童生徒が実際に解答した内容、得点等を解答結果データとして、コンテンツメタデータと分けている。

(2) アンケート設問の回答結果を連携する場合

図 4-8 にテスト・ドリルの解答結果のシステム連携の処理内容を示す。

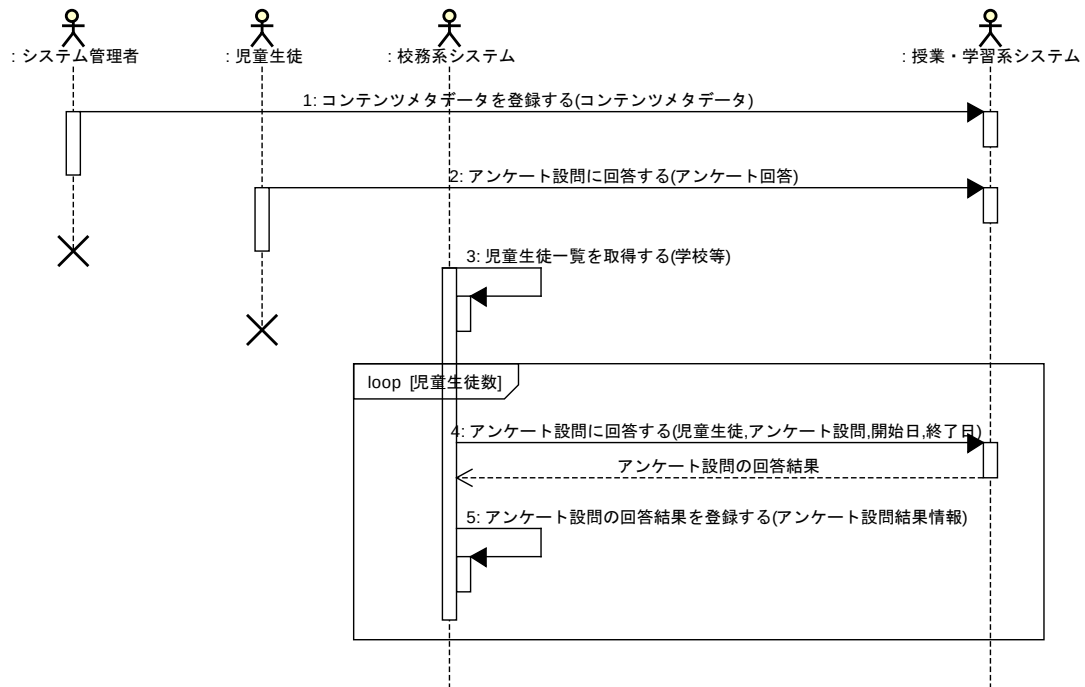


図 4-8 システム連携処理（児童生徒の学習理解度や自己認識状況の把握－アンケート）

以下、アンケート設問の回答結果のシステム連携の処理内容について説明する。

1. システム管理者は、授業・学習系システムに、アンケート設問のコンテンツメタデータを登録する。コンテンツメタデータは、スマートスクール・プラットフォーム技術仕様 API の連携対象外のため、事前に登録しておく必要がある。
2. 児童生徒は、授業・学習系システムにて、アンケート設問に回答する。回答結果は授業・学習系システムに回答結果として登録される。
3. 校務系システムは、自システムに登録されている児童生徒一覧を取得する。
4. 校務系システムは、3 で取得した児童生徒一覧より、各児童生徒のアンケート設問の回答結果を、児童生徒、対象のアンケート設問、日付期間を指定して、授業・学習系システムの学習記録情報 API サービスを呼び出し、取得する。
5. 校務系システムは、4 で取得した児童生徒のアンケート設問の回答結果を自システムに登録する。

4.2 システム間データ連携によるデータ活用事例

スマートスクール・プラットフォームでは、校務系システムと授業・学習系システムがデータ連携することにより、以下に示すデータ活用事例の実現を図ることを想定している。

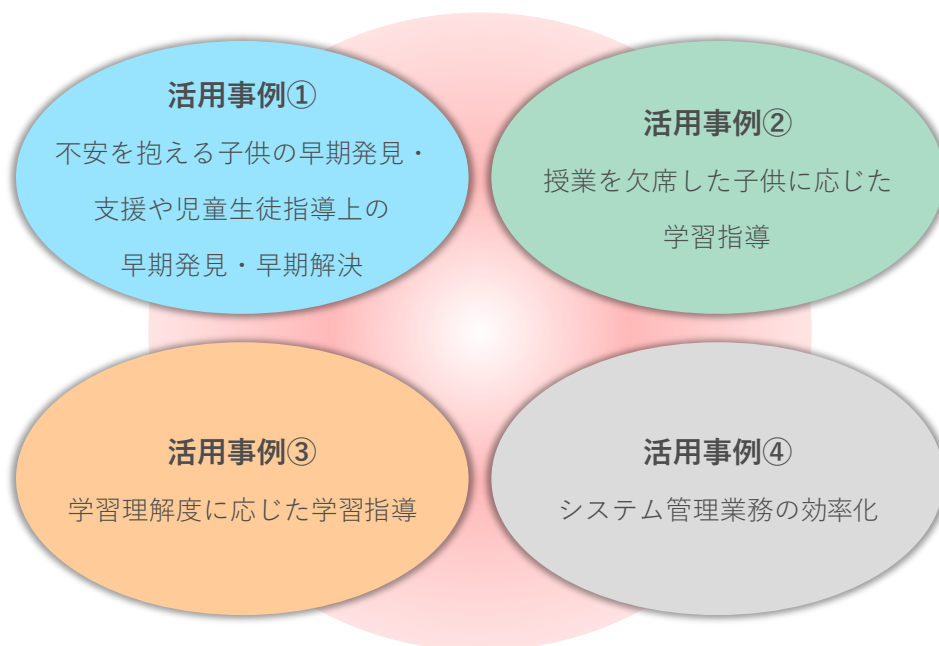


図 4-9 スマートスクール・プラットフォーム データ活用事例

上記活用事例①、②、③は教職員（主に学校管理職、担当教諭、養護教諭、スクールカウンセラー、スクールソーシャルワーカー）を対象にした活用事例であり、活用事例④は、校務系システム、授業学習系システムのシステム管理者を対象にしている。

それぞれのデータ活用事例では、文部科学省「エビデンスに基づいた学校教育の改善に向けた実証事業」のデータ活用パターンとの対応関係も示しておいた。データ活用パターン個々の具体的な事例については、文部科学省「エビデンスに基づいた学校教育の改善に向けた実証事業」（「5.3 参考文献」）を参照すること。

データ活用事例のユースケース図³を図 4-10 に示す。

³ ユーザ側の視点からシステムの機能、振る舞いを把握するための UML ダイアグラムのひとつ（UML については巻頭「用語集」を参照）。

図 4-10 では、データ活用事例（緑）が、スマートスクール・プラットフォーム技術仕様 API サービス（黄）との対応関係を図示している。

表 4-2 活用事例と連携対象データ

活用事例	連携対象となるデータ									
	児童生徒情報	教職員情報	学級情報	時間割情報	出欠情報	日常所見情報	保健室利用情報	テスト結果情報	ドリル結果情報	アンケート結果情報
【活用事例①】 不安を抱える子供の早期発見・支援や児童生徒指導上の早期発見・早期解決					●	●	●			●
【活用事例②】 授業を欠席した子供に応じた学習指導				●	●					
【活用事例③】 学習理解度に応じた学習指導								●	●	●
【活用事例④】 システム管理業務の効率化	●	●	●							

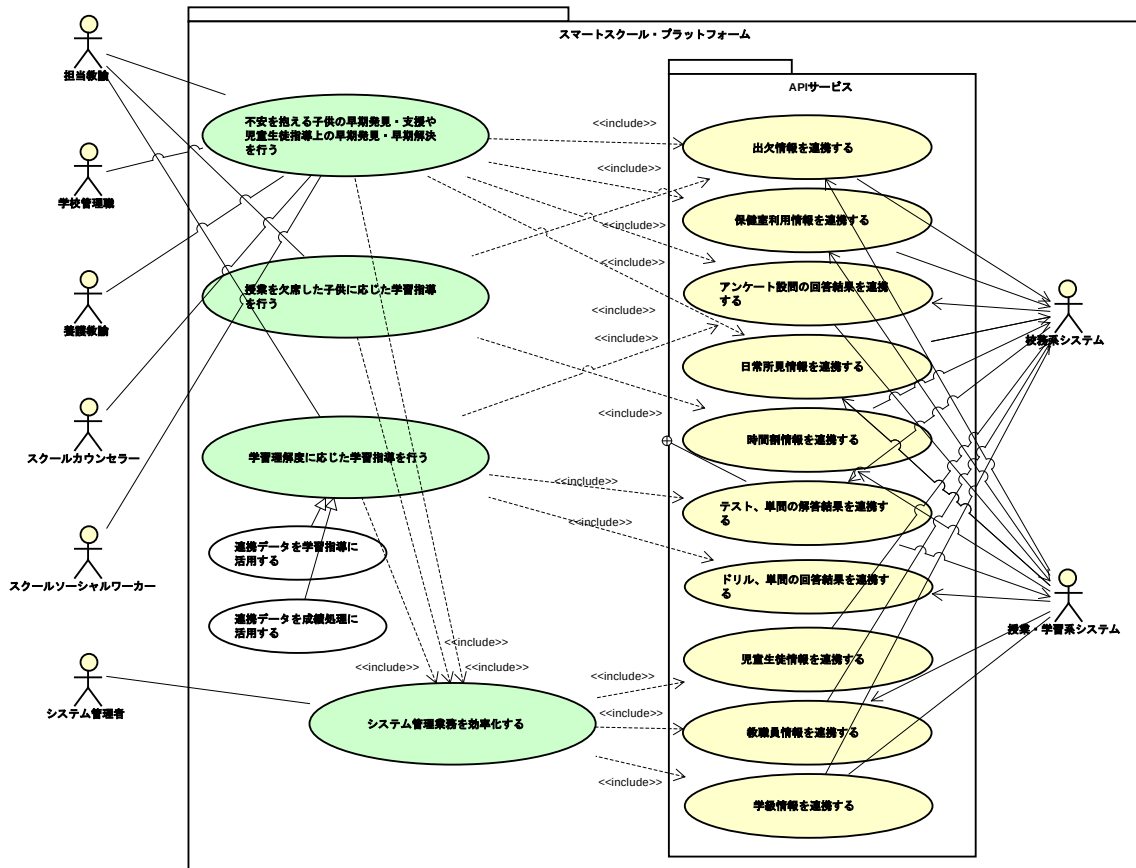


図 4-10 ユースケース図（データ活用事例）

以下、それぞれのデータ活用事例について解説する。

4.2.1 【活用事例①】不安を抱える子供の早期発見・支援や児童生徒指導上の早期発見・早期解決

本活用事例の概要を表 4-3 に示す。また、ユースケース図として図 4-11 に図示した。

表 4-3 活用事例①概要

項目	内容
ユースケース名	不安を抱える子供の早期発見・支援や児童生徒指導上の早期発見・早期解決を行う。
概要	下記、①②のいずれか ① 校務系システムの<u>児童生徒情報</u>、<u>教職員情報</u>、<u>出欠情報</u>、<u>保健室利用情報</u>、<u>日常所見情報</u>を授業・学習系システムに渡し、<u>アンケート回答結果</u>と併せて活用することで、不安を抱える児童生徒の把握や児童生徒指導上の注視すべき児童生徒を確認し適切な支援・解決を行う。 - 授業・学習系システム（教室等）において、連携データを分析・可視化し、不安を抱える子供の早期発見や支援、児童生徒指導上の早期発見・早期解決を支援する。 ② 授業・学習系システムの<u>アンケート回答結果</u>を校務系システムに渡し、<u>出欠情報</u>、<u>保健室利用情報</u>、<u>日常所見情報</u>と併せて活用することで、不安を抱える児童生徒の把握や児童生徒指導上の注視すべき児童生徒を確認し適切な支援・解決を行う。 - 校務系システム（職員室等）において、連携データを分析・可視化し、不安を抱える子供の早期発見・支援や児童生徒指導上の早期発見・早期解決を支援する。
目的	（対応する文科省データ活用パターン） 生徒指導の充実 - データに基づいた生活指導（担当教諭） - 児童生徒に対する組織的支援（養護教諭、スクールカウンセラー、スクールソーシャルワーカー） 学校経営の充実 - 管理職による児童生徒の把握（学校管理職）
アクター	- 学校管理職 - 担当教諭 - 養護教諭 - スクールカウンセラー - スクールソーシャルワーカー

本活用事例では、活用するデータとして、

- 児童生徒情報
- 教職員情報
- 出欠情報、保健室利用情報、日常所見情報
- 児童生徒アンケート回答結果（学習記録情報）

を想定している。

ここでは一例として、校務系システムに、児童生徒情報、教職員情報、出欠情報、保健室利用情報、日常所見情報が登録されていると想定し、これらの情報を、それぞれ、児童生徒情報 API サービス、教職員情報 API サービス、出欠情報 API サービス、保健室利用 API サービス、日常所見情報 API サービスにより、校務系システムから授業学習系システムにデータ連携させ、授業・学習系システム上での児童生徒アンケート回答結果と併せて、可視化、活用することを想定している。

また、もうひとつの一例として、授業・学習系システムに登録されている児童生徒アンケート回答結果を、学習記録情報 API サービスにより、校務系システムに連携させることにより、校務系システム側で同様の活用を図る。

本活用事例においては、上記データについてシステム連携をする必要があるが、システム連携の詳細については、

- 「4.1.3.2 児童生徒の転学等」（児童生徒情報の連携）
- 「4.1.3.3 教職員の異動」（教職員情報の連携）
- 「4.1.3.4 児童生徒の生活状況等の把握」（出欠情報、保健室利用情報、日常所見情報の連携）
- 「4.1.3.5 児童生徒の学習理解度や自己認識状況の把握」（児童生徒アンケート回答結果の連携）

に記載している。スマートスクール・プラットフォーム技術仕様 API により、校務系システム、授業・学習系システムが、日次で上記のデータ連携を行うことにより、本活用事例の実現を図る。

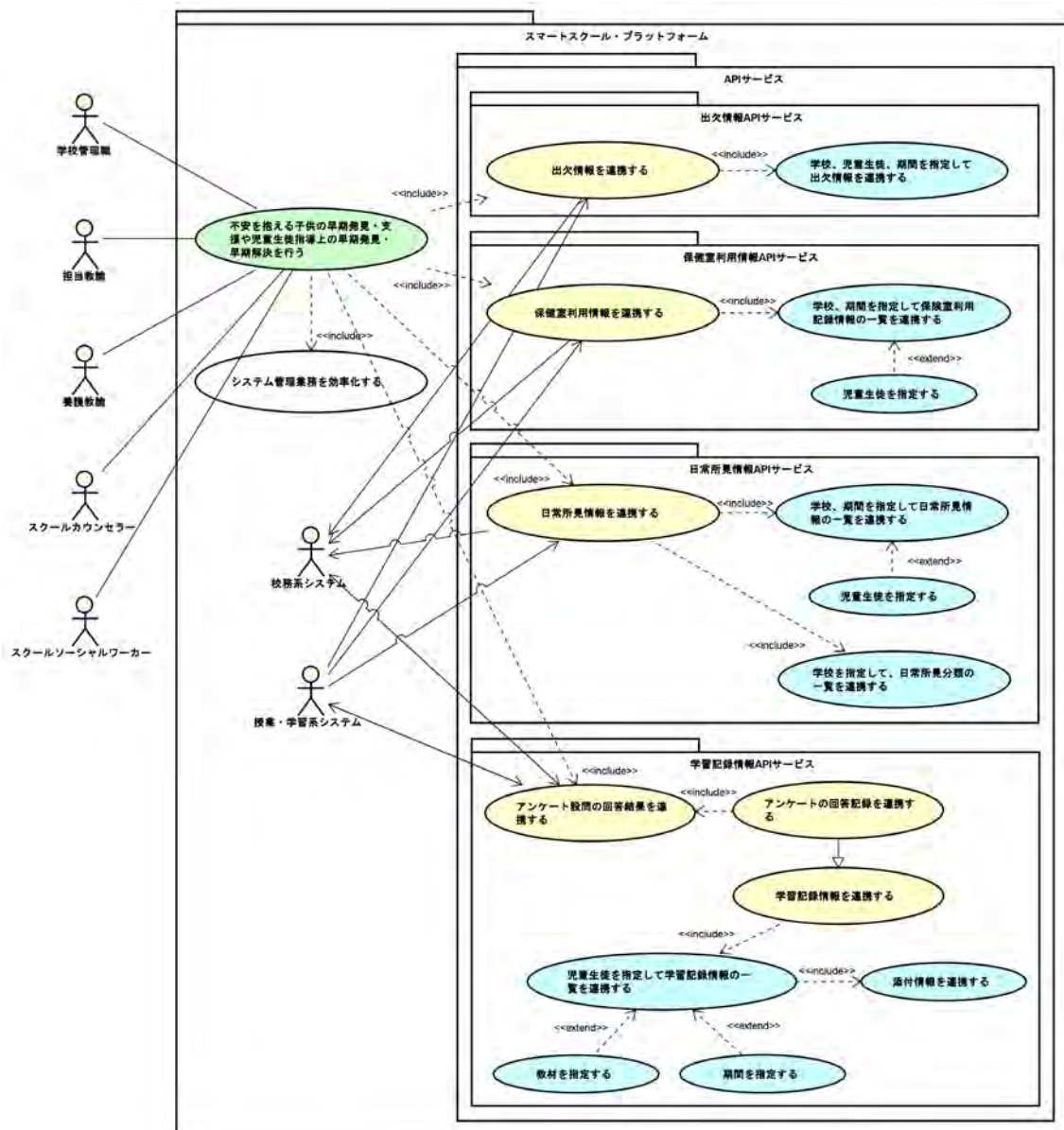


図 4-11 ユースケース図（活用事例①）

4.2.2 【活用事例②】授業を欠席した子供に応じた学習指導

本活用事例の概要を表 4-4 に示す。また、ユースケース図として図 4-12 に図示した。

表 4-4 活用事例②概要

項目	内容
ユースケース名	授業を欠席した子供に応じた学習指導を行う
概要	校務系システムの <u>出欠情報</u> 、 <u>時間割情報</u> を授業・学習系システムに渡し、活用して、児童生徒が欠席した日の時間割を元に、児童生徒に応じたドリル学習問題を課題提示する。 授業・学習系システム（教室等）において連携データを分析・可視化し、学習指導に活用する。
目的	（対応する文科省データ活用パターン） 学習指導の充実 一人ひとりに応じた学習指導
アクター	担当教諭

本活用事例では、活用するデータとして、

- 出欠情報
- 時間割情報

を想定している。

ここでは一例として、校務系システムに登録されている出欠情報、時間割情報を、授業・学習系システムへ連携することにより、児童生徒の授業の出欠状況と合わせた宿題等の課題の抽出、提示を行えるようにする。

本活用事例については、上記データについてシステム連携をする必要があるが、システム連携の詳細については、

- 「4.1.3.4 児童生徒の生活状況等の把握」（出欠情報の連携）
- 「4.1.3.1 時間割変更」（時間割情報の連携）

に記載している。スマートスクール・プラットフォーム技術仕様 API により、校務系システム、授業・学習系システムが、日次で上記のデータ連携を行うことにより、本活用事例の実現を図る。

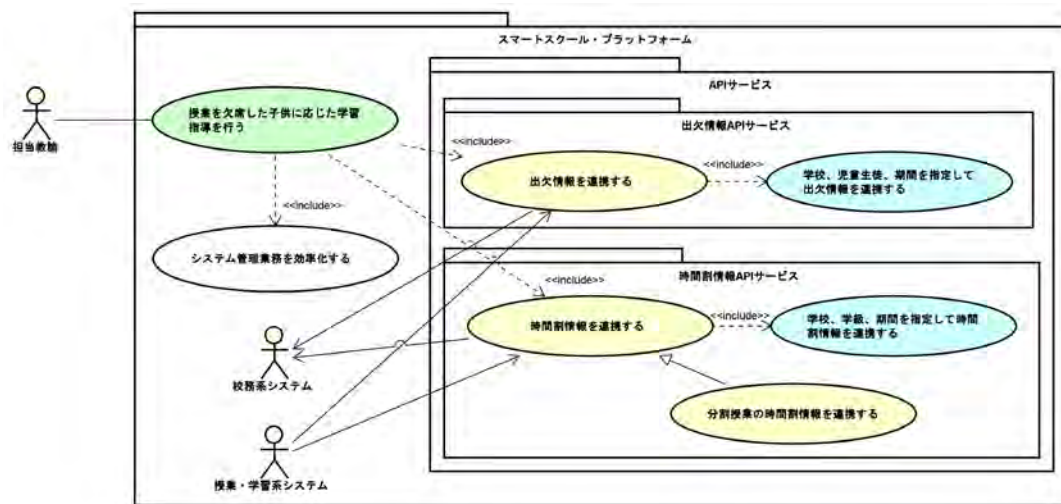


図 4-12 ユースケース図（活用事例②）

4.2.3 【活用事例③】学習理解度に応じた学習指導

本活用事例の概要を表 4-5 に示す。また、ユースケース図として図 4-13 に図示した。

表 4-5 活用事例③概要

項目	内容
ユースケース名	学習理解度に応じた学習指導を行う
概要	授業・学習系システムで児童生徒が実施した<u>学習結果</u>（テスト・ドリルの教科別・観点別の得点や単問ごとの正誤等）や<u>アンケート回答結果</u>を校務系システムに渡し、活用することで、学習支援が必要な児童生徒を把握し個に応じた指導や成績処理を行う。 - 校務系システム（職員室等）において連携データを分析・可視化し、学習指導に活用する。 - 校務系システム（職員室等）において連携データを分析・可視化し、成績処理に活用する。
目的	（対応する文科省データ活用パターン） 学習指導の充実 - 一人ひとりに応じた学習指導 - 授業改善
アクター	担当教諭

本活用事例では、活用するデータとして、

- テスト・ドリルの解答結果（学習記録情報）
- 児童アンケートの回答結果（学習記録情報）

を想定している。

ここでは一例として、上記データを授業・学習系システムから校務系システムへ連携させることにより、児童生徒の学習指導、成績評価への活用を図る。

本活用事例においては、上記データについてシステム連携をする必要があるが、システム連携の詳細については、

- 「4.1.3.5 児童生徒の学習理解度や自己認識状況の把握」（テスト・ドリルの解答結果、児童生徒アンケートの回答結果）

に記載している。スマートスクール・プラットフォーム技術仕様 API により、校務システム、授業・学習系システムが、日次で上記のデータ連携を行うことにより、本活用事例の実現を図る。

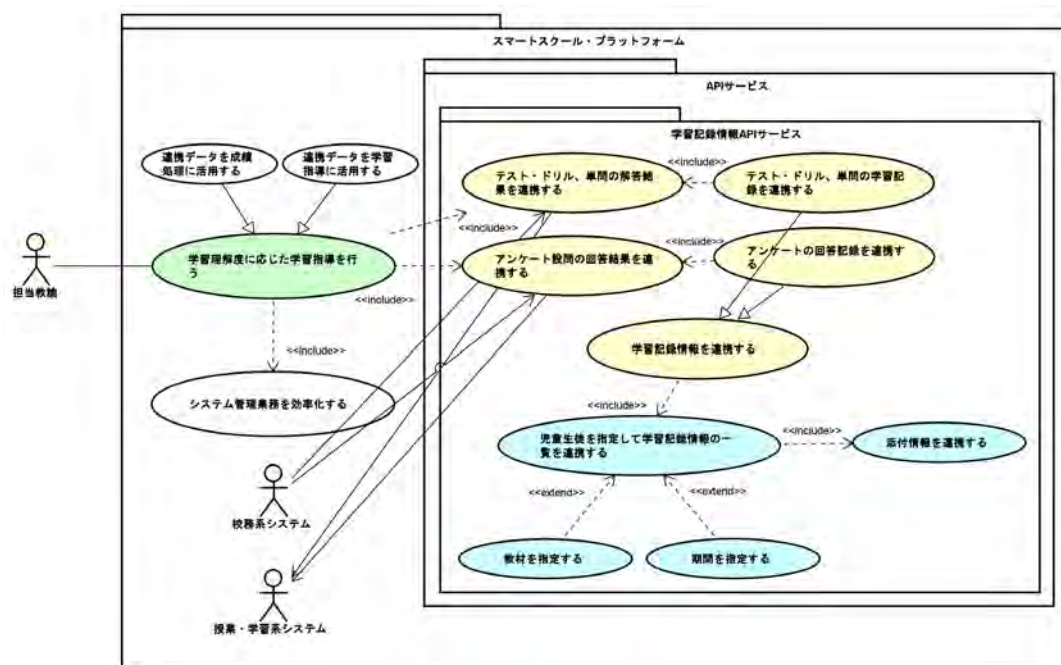


図 4-13 ユースケース図（活用事例③）

4.2.4 【活用事例④】システム管理業務の効率化

本活用事例の概要を表 4-6 に示す。また、ユースケース図として図 4-14 に図示した。

表 4-6 活用事例④概要

項目	内容
ユースケース名	システム管理業務を効率化する
概要	児童生徒の転学や学級編成等で、校務系システムに登録した <u>児童生徒情報</u> 、 <u>教職員情報</u> 、 <u>学級情報</u> を授業・学習系システムに渡すことで、システムごとの二重登録業務を削減する。 校務系データを授業・学習系システムにデータ連携し、業務を効率化する。（転入学等で発生する児童生徒情報等のシステム間同期）
目的	システム管理業務を効率化するため。
アクター	・ システム管理者

本活用事例では、活用するデータとして、

- 児童生徒情報
- 教職員情報
- 学級情報

を想定している。

ここでは一例として、校務系システムに登録されている児童生徒情報（属性情報、在学情報）、教職員情報、学級情報を、児童生徒の入学、卒業時など年度更新時や、日々の児童生徒の転学時、教職員の異動時に、児童生徒情報 API サービス、教職員情報 API サービス、学級情報 API サービスにより、校務系システムから授業・学習系システムにデータ連携させることで、システムの二重登録作業を不要にするなど、システム管理業務の効率化を図ることを想定している。

本活用事例においては、上記データについてシステム連携を必要があるが、システム連携の詳細については、

- 「4.1.2.1 児童生徒の入学・卒業等、学級編成、時間割作成」（児童生徒情報、学級情報の連携－年次処理）
- 「4.1.3.2 児童生徒の転学等」（児童生徒情報の連携－日次処理）
- 「4.1.3.3 教職員の異動」（教職員情報－日次処理）

に記載している。スマートスクール・プラットフォーム技術仕様 API により、校務システム、授業・学習系システムが、年次もしくは日次で上記のデータ連携を行うことにより、本活用事例の実現を図る。

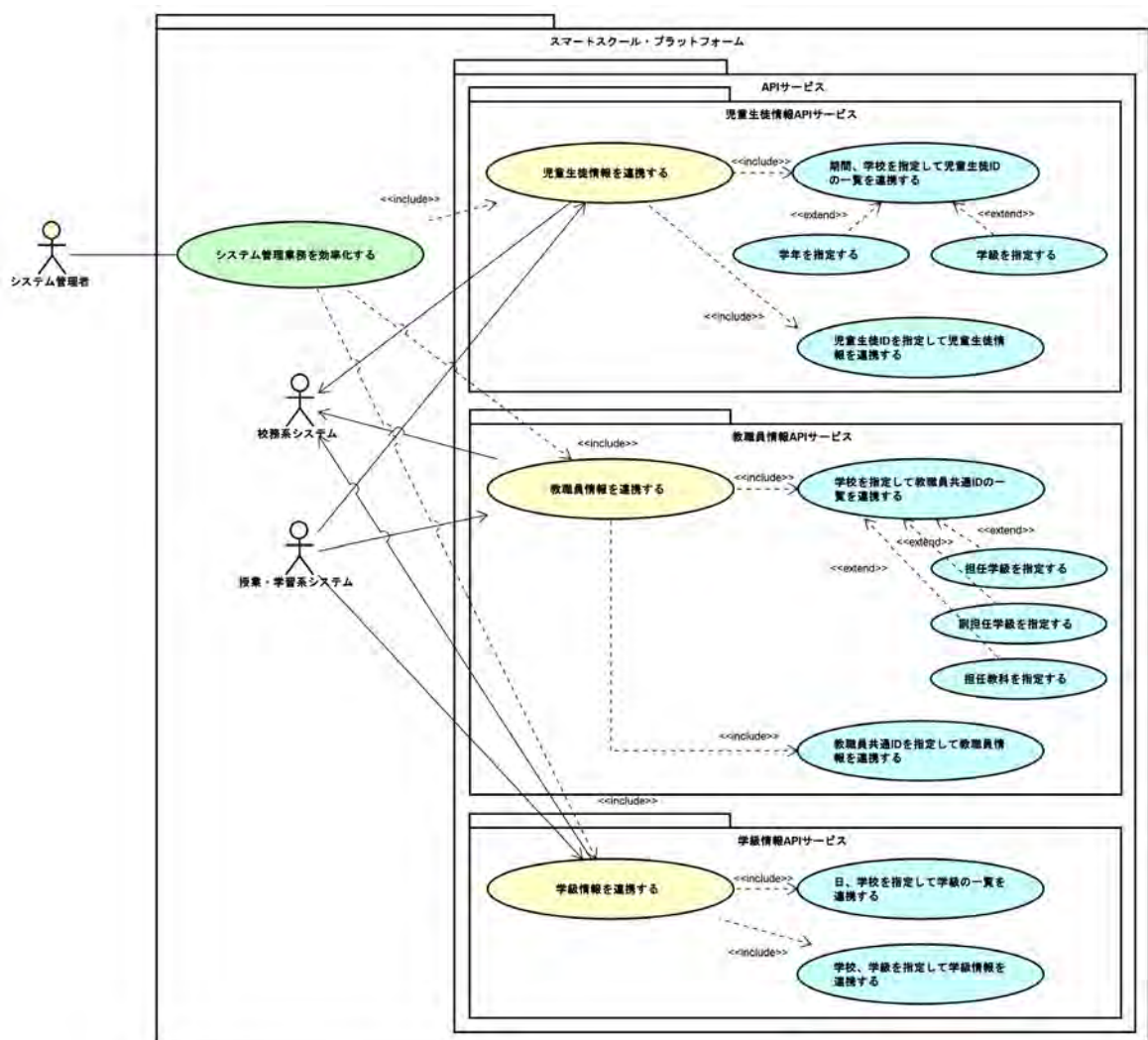


図 4-14 ユースケース図（活用事例④）

5. 卷末付録

5.1 補足資料

5.1.1 全体ユースケース

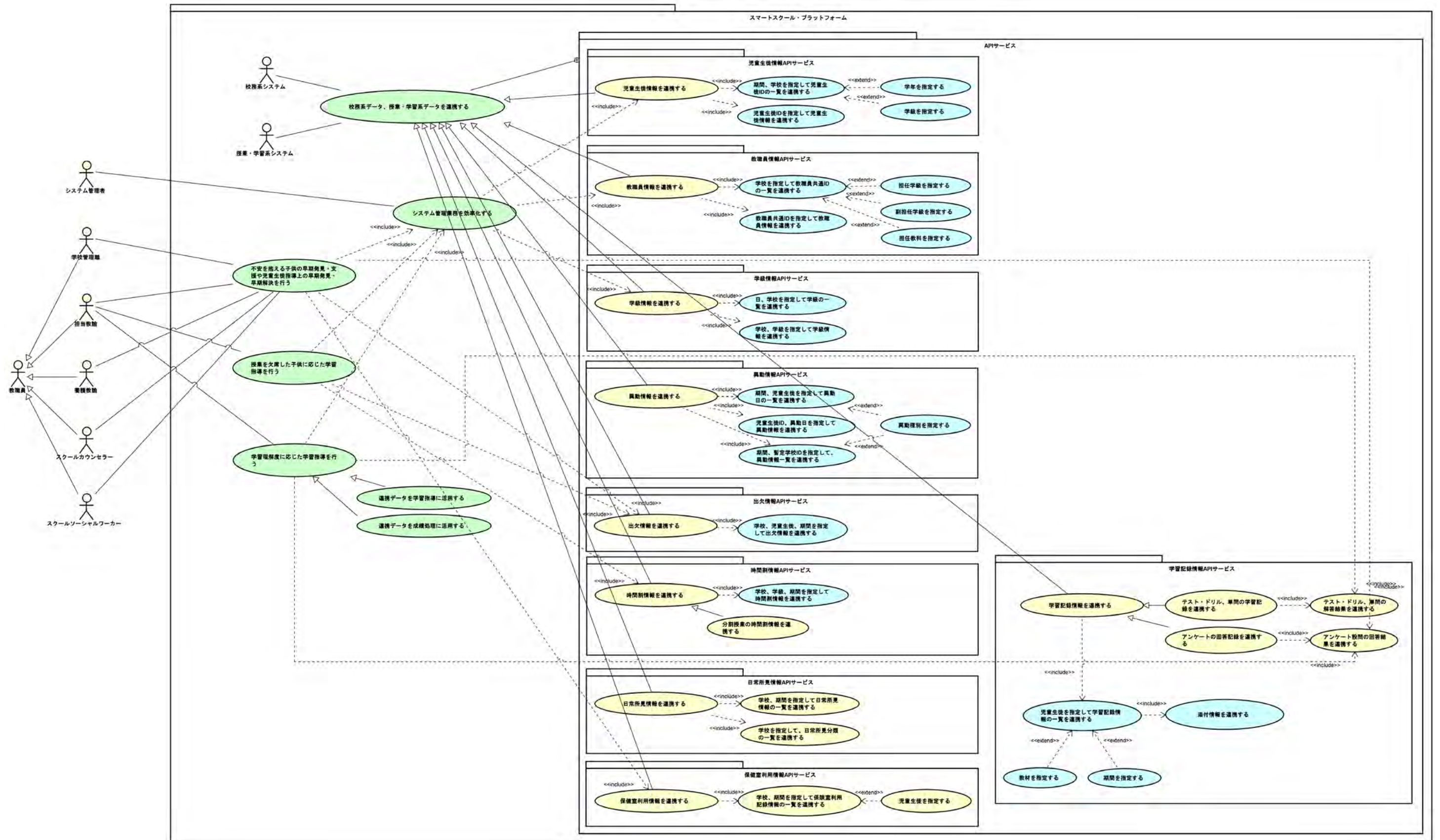


図 5-1 全体ユースケース

5.1.2 全体データモデル

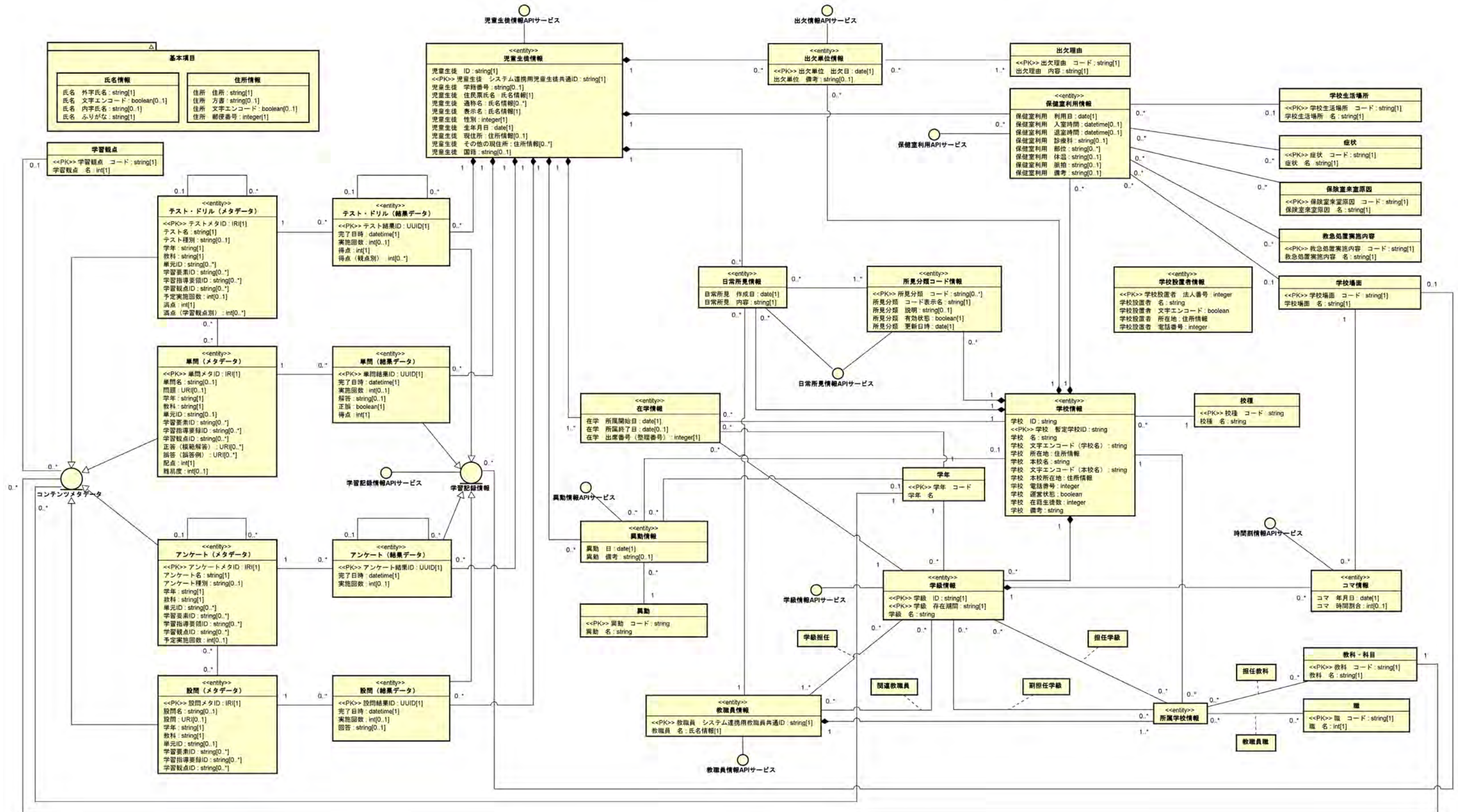


図 5-2 全体データモデル

5.1.3 全体データモデル（カテゴリ別）

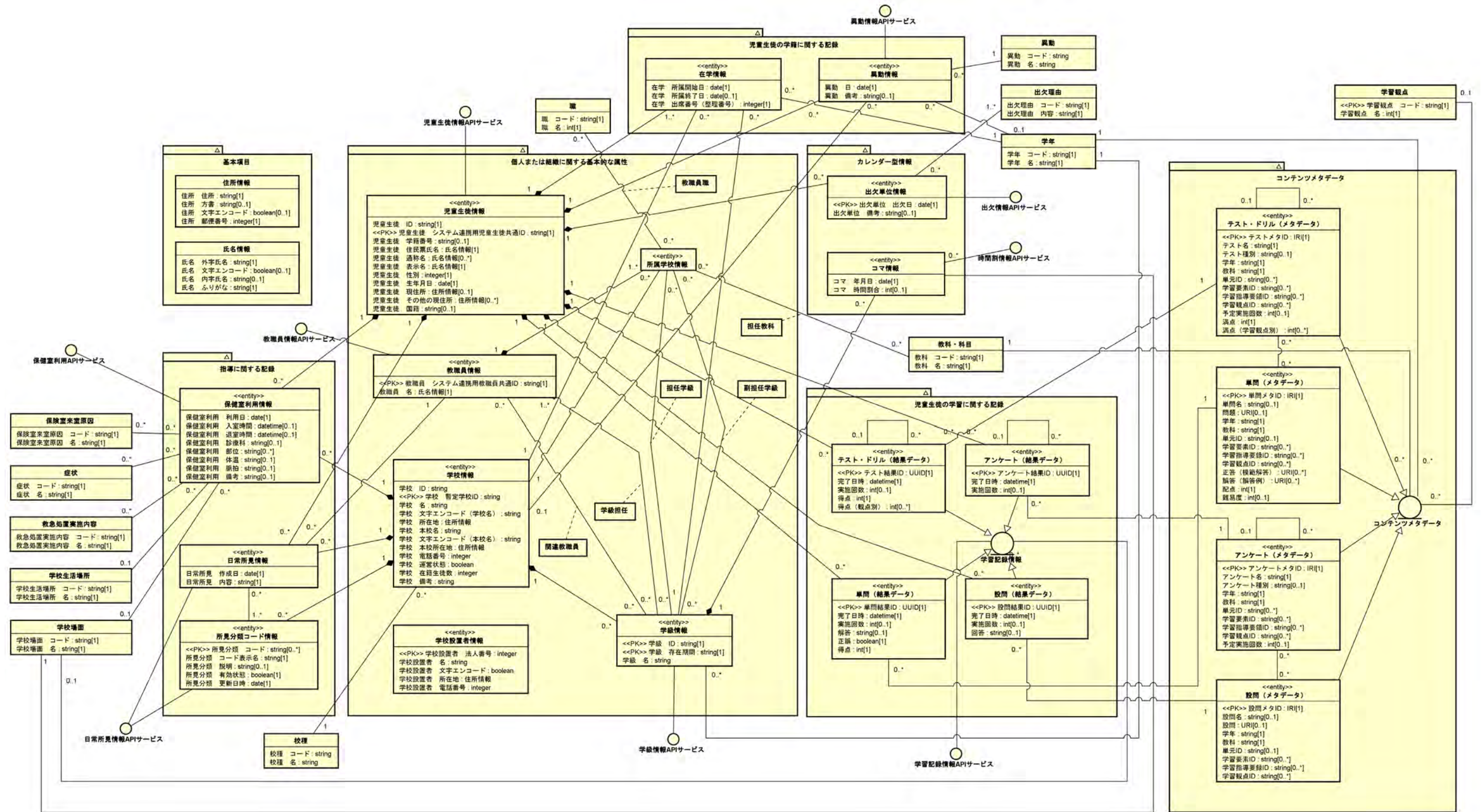


図 5-3 全体データモデル（カテゴリ別）

5.1.4 ステートメント全体構成

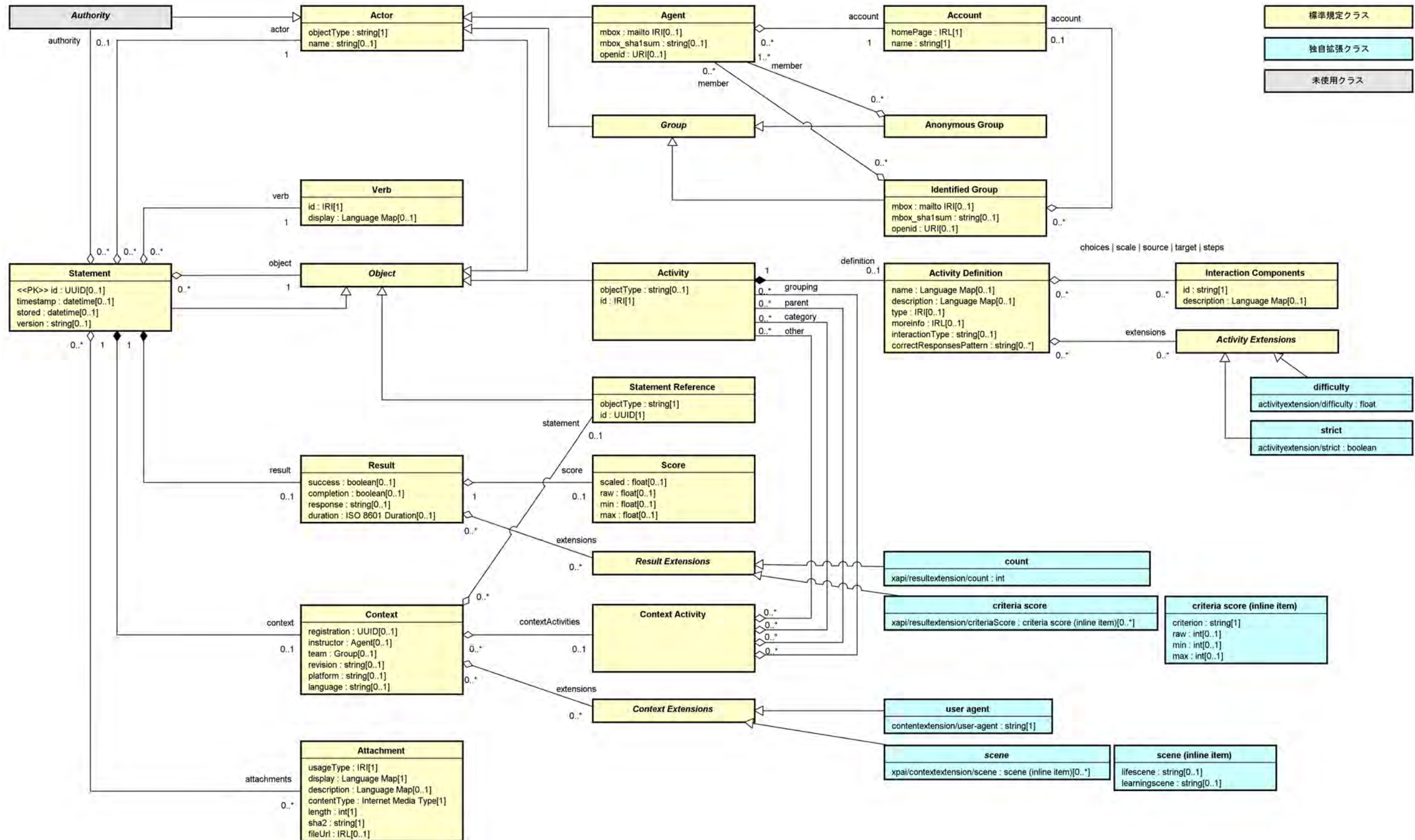


図 5-4 ステートメント全体構成

5.1.5 コンテンツメタデータ

5.1.5.1 テスト・ドリル

表 5-1 コンテンツメタデータ（テスト・ドリル）

項目名	最小	最大	値	項目説明
テストメタ ID	1	1	ユニーク ID	本テスト・ドリルのメタデータを特定する ID (Activity ID)
テスト名	1	1	自由記述	本テスト・ドリルの名称
テスト種別	1	1	コード値	本テスト・ドリルの種別 事前テスト、クイズ、小テスト、確認テスト、単元テスト、定期テスト、実力テスト、学力調査、自習ドリル
学年	1	1	コード値 (学年コード)	本テスト・ドリルの対象となる学年
教科	1	1	コード値 (教科・科目コード)	本テスト・ドリルの対象となる教科
単元 ID	0	n	コード値 ⁴	(予約項目) 本テスト・ドリルの対象となる単元 ID
学習要素 ID	0	n	コード値 ⁵	(予約項目) 本テスト・ドリルの対象となる学習要素 ID
学習指導要領 ID	0	n	コード値 ⁶	(予約項目) 本テスト・ドリルの対象となる学習指導要領 ID
学習観点 ID	0	n	コード値 (学習観点コード)	本テスト・ドリルの対象となる学習観点 ID
予定実施回数	0	1	数値	本テスト・ドリルを実施する予定（想定）回数
満点	1	1	数値	本テスト・ドリルの満点
満点（学習観点別）	0	n	数値	本テスト・ドリルの学習観点別の満点 学習観点 ID の数、並び順を合わせる
問題メタ ID	0	n		本テスト・ドリルに含まれる単問の単問メタ ID (Activity ID)
下位テストメタ ID	0	n		本テスト・ドリルに含まれる下位層のテストメタ ID (Activity ID)

⁴ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

⁵ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

⁶ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

5.1.5.2 単問

表 5-2 コンテンツメタデータ（単問）

項目名	最小	最大	値	項目説明
単問メタ ID	1	1	ユニーク ID	本問題のメタデータを特定する ID (Activity ID)
単問名	0	1	自由記述	本問題の名称
問題	0	1	URI	本問題の URI
学年	1	1	コード値	本問題の対象となる学年
教科	1	1	コード値	本問題の対象となる教科
単元 ID	0	1	コード値 ⁷	(予約項目) 本問題の対象となる単元 ID
学習要素 ID	0	n	コード値 ⁸	(予約項目) 本問題の対象となる学習要素 ID
学習指導要領 ID	0	n	コード値 ⁹	(予約項目) 本問題の対象となる学習指導要領 ID
学習観点 ID	0	n	コード値 (APPLIC の観点 ID)	本問題の対象となる学習観点 ID
正答 (模範解答)	0	n	URI	本問題の正答例の URI
誤答 (誤答例)	0	n	URI	本問題の誤答例の URI
配点	1	1	数値	本問題の配点
難易度	0	1	コード値	本問題の難易度。5 段階で表現する。

⁷ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

⁸ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

⁹ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

5.1.5.3 アンケート

表 5-3 コンテンツメタデータ（アンケート）

項目名	最小	最大	値	項目説明
アンケートメタ ID	1	1	ユニーク ID	本アンケートのメタデータを特定する ID (Activity ID)
アンケート名	1	1	自由記述	本アンケートの名称
アンケート種別	1	1	コード値	本アンケートの種別
学年	1	1	コード値 (学年コード)	本アンケートの対象となる学年 学年によらないアンケートは「n/a」を設定
教科	1	1	コード値 (教科・科目コード)	本アンケートの対象となる教科 教科によらないアンケートは「n/a」を設定
単元 ID	0	n	コード値 ¹⁰	(予約項目) 本アンケートの対象となる単元 ID
学習要素 ID	0	n	コード値 ¹¹	(予約項目) 本アンケートの対象となる学習要素 ID
学習指導要領 ID	0	n	コード値 ¹²	(予約項目) 本アンケートの対象となる学習指導要領 ID
学習観点 ID	0	n	コード値 (学習観点コード)	本アンケートの対象となる学習観点 ID
予定実施回数	0	1	数値	本アンケートを実施する予定 (想定) 回数
設問メタ ID	0	n		本アンケートに含まれる設問の設問メタ ID (Activity ID)
下位アンケートメタ ID	0	n		本テスト・ドリルに含まれる下位層のアンケートメタ ID (Activity ID)

5.1.5.4 設問

表 5-4 コンテンツメタデータ（設問）

項目名	最小	最大	値	項目説明
設問メタ ID	1	1	ユニーク ID	本設問のメタデータを特定する ID (Activity ID)
設問名	0	1	自由記述	本設問の名称
設問	0	1	URI	本設問の URI
学年	1	1	コード値 (学年コード)	本設問の対象となる学年 学年によらない設問は「n/a」を設定
教科	1	1	コード値 (教科・科目コード)	本設問の対象となる教科 教科によらない設問は「n/a」を設定
単元 ID	0	1	コード値 ¹³	(予約項目) 本設問の対象となる単元 ID
学習要素 ID	0	n	コード値 ¹⁴	(予約項目) 本設問の対象となる学習要素 ID
学習指導要領 ID	0	n	コード値 ¹⁵	(予約項目) 本設問の対象となる学習指導要領 ID
学習観点 ID	0	n	コード値 (教科・科目コード)	本設問の対象となる学習観点 ID

¹⁰ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

¹¹ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

¹² コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

¹³ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

¹⁴ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

¹⁵ コード体系は規定しない。本項目は将来的な利用を想定した予約項目である。

5.1.6 結果データ

5.1.6.1 テスト・ドリル

表 5-5 結果データ（テスト・ドリル）

項目名	最小	最大	値	項目説明
テスト結果 ID	1	1	ユニーク ID	テスト・ドリルの結果データを特定する ID (Statement ID)
テストメタ ID	1	1		本テスト結果のテストメタ ID (Activity ID)
完了日時	1	1	日時	児童生徒が本テスト・ドリルを完了した日時
学校生活場面	0	1	コード値 (学校場面コード)	児童生徒が本テスト・ドリルを実施した学校生活場面
実施回数	0	1	数値	児童生徒が本テスト・ドリルを実施した回数
得点	1	1	数値	児童生徒が本テスト・ドリルで得た点数
得点（観点別）	0	n	数値	児童生徒が本テスト・ドリルで得た学習観点別の点数 メタデータの学習観点 ID の数、並び順を合わせる
児童生徒 ID	1	1	文字列	本テスト・ドリルに解答した児童生徒

5.1.6.2 単問

表 5-6 結果データ（単問）

項目名	最小	最大	値	項目説明
単問結果 ID	1	1	ユニーク ID	本問題の結果データを特定する ID (Statement ID)
単問メタ ID	1	1		本問題結果の問題メタ ID (Activity ID)
完了日時	1	1	日時	児童生徒が本問題の解答を完了した日時
学校生活場面	0	n	コード値 (学校場面コード)	児童生徒が本問題を解答した学校生活場面
実施回数	0	1	数値	児童生徒が本問題を解答した回数
解答	0	1	その他	児童生徒が本問題を解答した内容 数値、文字列の場合、string として response に入れる 上記以外の場合、Attachments を使い、ファイル添付
正誤	1	1	その他	児童生徒が本問題を解答した正誤判定
得点	1	1	数値	児童生徒が本問題を解答し得た得点
児童生徒 ID	1	1	文字列	本問題に解答した児童生徒

5.1.6.3 アンケート ¹⁶

表 5-7 結果データ（アンケート） ¹⁷

項目名	最小	最大	値	項目説明
アンケート結果 ID	1	1	ユニーク ID	アンケートの結果データを特定する ID (Statement ID)
アンケートメタ ID	1	1		本アンケート結果のアンケートメタ ID (Activity ID)
完了日時	1	1	日時	児童生徒が本アンケートを完了した日時
学校生活場面	0	1	コード値 (学校場面コード)	児童生徒が本アンケートを実施した学校生活場面
実施回数	0	1	数値	児童生徒が本アンケートを実施した回数
児童生徒 ID	1	1	文字列	本アンケートに回答した児童生徒

5.1.6.4 設問

表 5-8 結果データ（設問）

項目名	最小	最大	値	項目説明
設問結果 ID	1	1	ユニーク ID	本設問の結果データを特定する ID (Statement ID)
設問メタ ID	1	1		本設問結果の設問メタ ID (Activity ID)
完了日時	1	1	日時	児童生徒が本設問の回答を完了した日時
学校生活場面	0	n	コード値 (学校場面コード)	児童生徒が本設問を回答した学校生活場面
実施回数	0	1	数値	児童生徒が本設問を回答した回数
回答	0	1	その他	児童生徒が本設問を回答した内容 数値、文字列の場合、string として response に入れる 上記以外の場合、Attachments を使い、ファイル添付
児童生徒 ID	1	1	文字列	本設問に回答した児童生徒

¹⁶ 未使用

¹⁷ 未使用

5.2 参考資料

5.2.1 全体 ER モデル（論理モデル）

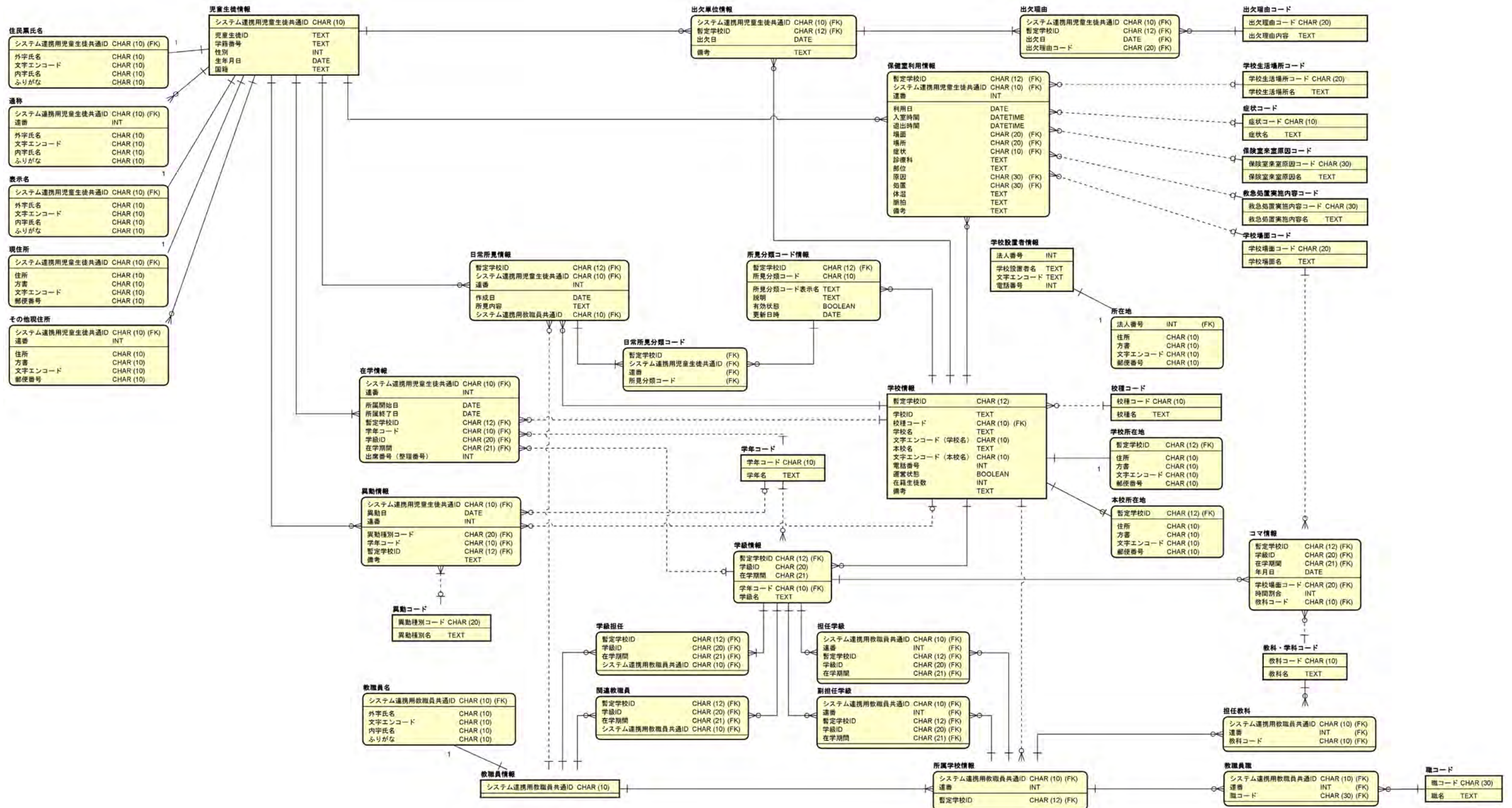


図 5-5 全体 ER モデル（論理モデル）

5.2.2 全体 ER モデル（物理モデル）

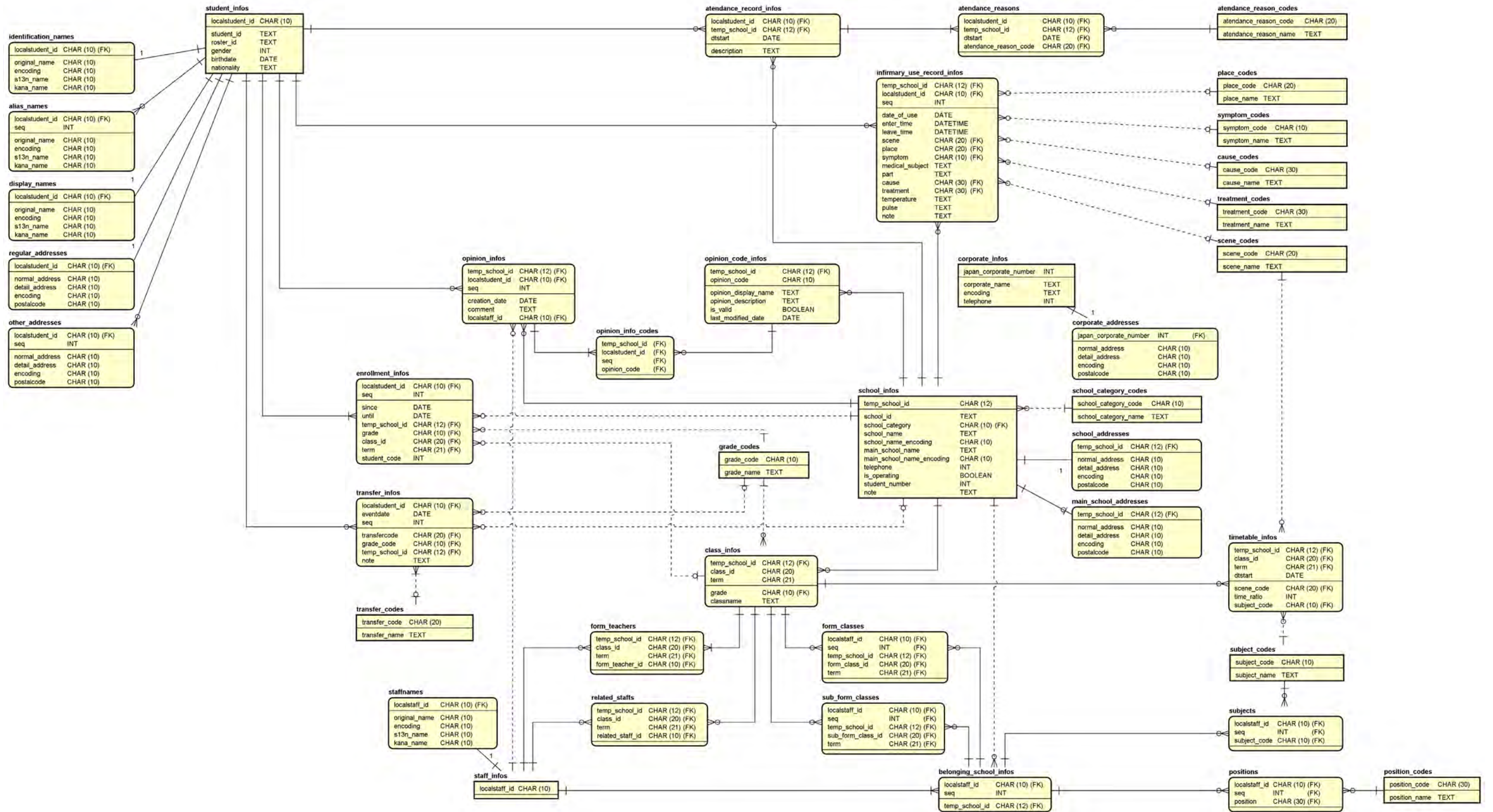


図 5-6 全体 ER モデル（物理モデル）

5.3 参考文献

- エビデンスに基づいた学校教育の改善に向けた実証事業¹⁸ – 文部科学省
http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1387543.htm
- 教育情報セキュリティポリシーガイドライン – 文部科学省
https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1397369.htm
- UML - OMG : Object Management Group
<https://www.uml.org/>
- jCal – RFC7265
<https://tools.ietf.org/html/rfc7265>
- xAPI – 日本語版
https://github.com/elc-gh/xAPI-Spec_ja/blob/master/xAPI_ja.md
- xAPI – 原文（英語）
<https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md>

¹⁸ 平成 31 年度（令和元年度）から、「次世代学校支援モデル構築事業」（平成 29 年度から同 30 年度）を基に、事業名称が「エビデンスに基づいた学校教育の改善に向けた実証事業」に変更された。